Query Refinement based on Comprehensive Representation of Multiple Topics

（複数トピックの包括的提示による検索支援に関する研究）

Hiromi Wakaki

December, 2006

The University of Tokyo
(Supervisor: Jun Adachi)

Abstract

In this thesis, I propose a method for supporting query refinement by using term clusters of topical terms, i.e., terms closely related to a specific topic, extracted from a retrieved set of documents. First, a new hypothesis is proposed that a term co-occurring frequently with a limited number of terms in a retrieved document set can establish a distinct topic in the document set. Next, three formulae of term-weighting are established based on the aforementioned hypothesis. Then, I examine whether these formulae can extract appropriate terms for query refinement by comparing them with other existing term-weighting methods. After that, I generate term clusters by using those terms. I also examine the quality of term clusters by checking whether the clusters can support query refinement. Additionally, to evaluate objectively and quantitatively whether a term-weighting method can extract topical terms, I propose two new tools. After ensuring the suitableness of these tools for measuring topic partiality, I use them to evaluate the performance of my formula in comparison with the other existing methods. The results show that my method works well. Finally, I describe a new demonstration system that embodies the proposed method to show topical term clusters which can be used for query refinement.

When we use existing search engines, we enter only a few terms to form a query. Even if we use effective query terms, e.g., proper nouns and technical terms, such a short query is likely to be ambiguous. As a result, we must select the documents of interest from a large number of retrieved documents which may have a wide variety of content. I propose a method for supporting query refinement by using term clusters of topical terms extracted from a retrieved set of documents.

I assume that a topic is implied by a specific set of terms that frequently co-occur in the same documents. Here, I introduce a new hypothesis of term importance called "*Tangibility*". A term is said to have Tangibility when it frequently co-occurs exclusively with a specific set of terms.

Existing methods for term extraction aim to extract terms corresponding to the dominant topic of a given document set. However, because such terms likely co-occur with a wide variety of other terms, we cannot discriminate topics. In contrast, my method aims to extract terms exclusively related to one of those topics.

I propose three new term-weighting methods, i.e., TNG1, TNG2, and TNG, based on Tangibility. Using the NTCIR3 Web Retrieval Task, I examined the performances of TNG1 and TNG2 in comparison with the other existing term-weighting methods, e.g., Mutual Information (MI), Kullback-Leibler Divergence (KLD), and Robertson's Selection Value (RSV). I tested whether the proposed methods can narrow the search by using terms from the searched documents to replace the user-provided query terms as a semiautomatic query expansion. In this experiment, TNG1, TNG2, and RSV significantly increased overall precision, and TNG2 achieved the best overall average precision. The most important finding is that TNG1 and TNG2 showed qualitative differences from RSV. While RSV tends to extract terms having general meanings, TNG1 and TNG2 can extract many technical terms used in specific domains pertaining to the query.

TNG is an improvement of TNG1 and TNG2. First, the following equation measures how much the probability of term t_j's appearance increases by adding the condition that term t_i appears.

$$\Delta_{t_i}(t_j) \quad = \quad P(t_j|t_i) \times \log \frac{P(t_j|t_i)}{P(t_j)} \tag{1}$$

By setting $F_i = \{t_j|\Delta_{t_i}(t_j) > 0\}$, the formula for TNG is obtained:

$$TNG(t_i) = \frac{\sum_{t_k \in F_i}(\Delta_{t_i}(t_k))}{|F_i|} \tag{2}$$

A far as I know, there are no data sets with the correct topic labels assigned to each term. However, since my method aims to generate term clusters, each of which corresponds to a topic, I need to conduct experiments to test whether a term is strongly related to one of the topics contained in a document set. Therefore, to evaluate objectively and quantitatively whether a term-weighting method can extract topical terms, I propose two new tools; Topic Label and Topical Skewness. Topic Label indicates the topic to which a given term is most closely related. Topical Skewness shows how exclusively a given term relates to one topic. When the experimental data set has document

categories, we can assign a Topic Label to every term and compute Topical Skewness for every term. In the experiment, I used document sets, each of which were generated by mixing three document categories. In addition, I used the category names as the labels assigned by the Topic Label. Here, by collecting data from subjective evaluations of human evaluators, I tested whether the Topic Labels given by the tools corresponded to the labels given by the evaluators. I also checked whether the Topical Skewness of each term correlated with the number of people who gave a term the same label as its Topic Label. The results show that these tools are suitable for evaluating topic separation. Therefore, I used Topic Label and Topical Skewness to compare TNG with other term-weighting methods.

I examined the quality of extracted terms by checking whether each term is strongly related to one topic. I compared TNG with other existing term weighting methods on seven data sets, i.e., NTCIR3, NTCIR4, NTCIR-CLIR, Sankei Sports News, Dmoz, Reuters and Newsgroup20. All the data sets except Sankei Sports News are commonly used. The data set for document classification surely had categories. Moreover, a test collection for the evaluation of information retrieval accompanies relevant document sets for each test query, so I used both types of data. Moreover, three of the largest categories were mixed as pseudo-data including multiple topics. The results of this experiment show that TNG can extract terms strongly related to any one of several topics contained in the document set. I divided the extracted terms into clusters by using a distributional clustering algorithm, which leads to agglomerates of terms frequently co-occurring with each other. With respect to the average precision of documents retrieved by the clusters, TNG outperforms other existing methods, e.g., MI, KLD, and so on. Furthermore, TNG has good completeness of categories of documents retrieved by the term clusters.

I also examined how TNG compared with MI and RSV by using human subjective evaluations. The results also show that TNG can extract terms strongly related to any one topic. I thus conclude that TNG is an efficient term weighting method for detecting topics included in a heterogeneous set of documents. I also examined the quality of term clusters by checking whether the clusters can support query refinement. I first extracted topical terms from a synthesized heterogeneous document set, which is a surrogate for a document set retrieved by a real search engine. Then I constructed clusters of the extracted terms and used each term cluster as a query to assign a ranking to the documents by using the probabilistic information retrieval model. I used two evaluation criteria:

concentration and completeness. First, when the top-ranked documents given by each term cluster relate to the same topic, a topic-focusing search can be realized by using any one of the constructed clusters. Second, when different term clusters provide different sets of top-ranked documents, a search covering a wide variety of topics can be realized by using all term clusters. I evaluated TNG with these two criteria, and TNG outperformed other existing methods in most cases.

I developed a system using TNG to show topical term clusters that can be used for query refinement. My system is implemented in the Ruby programming language, CGI in Perl, and wget. I used MeCab as the part-of-speech (POS) and morphological analyzer for Japanese. First, this system gathers the top 500 URLs as ranked by Google for a query given by the user. Then it removes the dead links and the URLs referring to non-HTML documents. By removing HTML tags, I get the raw text of each document. As the downloaded documents show a wide divergence in their lengths, I extract snippets (i.e., short summaries of documents) of constant length from each document. The system calculates TNG scores by regarding each snippet as a single document and generates term clusters with my improved distributional clustering algorithm. The user can use each term cluster as additional query terms to expand the original query. I also examined TNG in comparison with Clusty.

I think that query ambiguity is caused by at least the following two reasons. The first is polysemy, i.e, the multiplicity of query meanings. The second reason is the multiplicity of the perspectives, i.e., facets, from which we view the concept or the object referred to by the query. As a result of the experiments, I observed the following two solutions. As for the polysemy, we can use the term clusters generated by my system to obtain distinct search results relating to distinct multiple meanings of the original query. As for the facets, I believe that the term clusters that my system provides are good candidates for the facets to be organized in a hierarchical structure.

Acknowledgements

I would like to gratefully acknowledge the enthusiastic supervision and endless encouragement of my adviser Prof. Jun Adachi during this work. I would also like to thank Prof. Jun Adachi for providing me with excellent facilities to pursue my work, and for ensuring financial support throughout my studies. I would like to thank Prof. Atsuhiro Takasu for the technical discussions on my work and great support. I would like to acknowledge the help of my senior colleague, Dr. Tomonari Masada for his enormous support and encouragement.

I am also grateful to the members of my committee for taking the time to guide me through my dissertation. I would like express my appreciation to Prof. Masaru Kitsuregawa, Prof. Mitsuru Ishizuka, Prof. Toru Asami, Prof. Hitoshi Aida, and Prof Nobuaki Minematsu for their review of the thesis, and their valuable comments, discussions to my work. I am also grateful to my previous adviser Prof. Hitoshi Iba for the discussions, encouragement and support.

I would like to thank Dr. Yutaka Matsuo, Dr. Naohiro Matsumura, Dr. Takashi Ninomiya, Dr. Yasuhiro Akagi, Dr. Yutaka Inoue, Dr. Ryota Ozaki, and Naoyuki Okazaki for their technical discussions on my work. I am also grateful to my colleagues and friends at Adachi Laboratory, Ishizuka Laboratory, and Asano Laboratory in the University of Tokyo for their valuable discussions and suggestions.

Finally, I am forever indebted to my parents and my grandparents for their understanding, encouragement, and enormous support.

Hiromi Wakaki

The University of Tokyo

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

When we use existing search engines such as Google*, Yahoo† and MSN‡, we enter only a few terms to form a query [JSBS98][one]. Then the search engines often return a long list of search results. Even if we use effective query terms, e.g., proper nouns and technical terms, various topics related to the query terms can be contained in the search results retrieved by such a short query. Therefore, we must select the documents we are interested in from the list by examining the titles and snippets. This is a time-consuming task because the list is unstructured, and it is not easy for web users to understand the multiple topics contained in the search results.

For example, assume that you must go to Amsterdam for business trip but know nothing about the city. Travel guide books for Amsterdam present well-organized information though it is limited in amount. In contrast, the information given by search engines provides far more information about Amsterdam though it is not well-organized. Guide books provide a comprehensive representation of the table of contents, and you will find multiple topics with guide books. In general, however, you cannot always use guides when searching for information. As a result, you will add some topical terms to "Amsterdam" and search web again and again. It is not easy for web users to come up with appropriate terms to be added and to cover multiple topics, e.g. sightseeing, flight ticket, history, and climate. A possible solution to this problem is a comprehensive representation of multiple topics

*http://www.google.com
†http://www.yahoo.com
‡http://search.msn.com

related to the query.

In this study, I propose a method for supporting query refinement by using clusters of topical terms extracted from a retrieved set of documents. I assume that a topic is implied by a specific set of terms that frequently co-occur in the same documents. Therefore, I introduce a new measure of term importance called *tangibility*. A term is said to have *tangibility* when it frequently co-occurs exclusively with a specific set of terms. Existing methods for term extraction aim to extract terms corresponding to the dominant topic of a given document set, e.g., travel, Europe, and Netherlands. However, because such terms are likely to co-occur with a wide variety of other terms, we cannot distinguish isolated topics. In contrast, my method aims to extract terms exclusively related to one of those topics, e.g., Rembrandt, Schiphol Airport, and canals. Then, I divide the extracted terms into clusters using a distributional clustering algorithm, which leads to agglomerates of terms frequently co-occurring with each other.

First, I will discribe my experiment for initial two formulae of *tangility*, i.e., TNG1 and TNG2. The results show that my method is successful in discovering terms, which are extracted by TNG1 and TNG2, that can be used to narrow the search. Moreover, we acquire new knowledge specialized in the context of the query if we did not know one or more of the terms presented by my method. Next, I will describe my experiment for improved formula of tangibility, i.e., TNG. My method tries to extract terms exclusively related to one of those topics. Then, I divide the terms extracted by TNG into clusters by using improved distributional clustering algorithm, which leads to agglomerates of terms frequently co-occurring with each other. The result shows term clusters that individually correspond to one topic contributes to the discovery of good terms for query expansion. When some of the terms contained in the clusters are unfamiliar with the user, they can be used for learning support. Finally, I will describe my system using proposed TNG and generating term clusters to show topical term clusters that can be used for query refinement. I also subjectively examined the performance of TNG in comparison with Clusty, which is Vivisimo's meta-search engine that offers enhanced features such as clustering. The result shows that the term clusters generated by my system using TNG can distinguish the topics of the search. Moreover, the term clusters help us to find multiple meanings of the query and to discover unexpected topics related to it.

I think that query ambiguity is caused by at least the following two reasons. The first is polysemy, i.e, the multiplicity of query meanings. When queries consists of a small number of terms,

they are likely to refer to multiple concepts or multiple objects. The second reason is the multiplicity of the perspectives from which we view the concept or the object referred to by the query. We call these perspectives *facets*. Even when a query refers to a unique concept or a unique object, we can view the concept or the object from various perspectives. Therefore, the documents retrieved by such a query may provide multiple perspectives from which we can view the concept or the object referred to by the query. I aim to observe the solution for these ambiguities with the ultimate objective of this study.

1.2 Organization of This Paper

The rest of the thesis is organized as follows. Chapter 2 introduces related works ; Chapter 3 proposes my method; Chapter 4 reports the details of the experiments for proposed method; Chapter 5 shows my system using proposed method and reports the details of the experiments for the system; and Chapter 6 concludes with a summary of the paper.

Chapter 2

Related Work

2.1 Overview

Organizing Search Results Result categorization and query refinement are well-known techniques for improving search results. For the purpose of result categorization, clustering and classification are used in the categorization algorithms [CKPT92] [HP96] [PF00] [ZE99]. Scatter/Gather [CKPT92] [HP96] is a system in which the clustering approach was tested. On the other hand, DynaCat [PF00] is a prototype system that uses the classification approach. Both of them aim to categorize documents retrieved as search results. In this thesis, I aim to make term clusters corresponding to each topic contained in the search results. I make term clusters rather than document clusters, and each term cluster focuses on one of the topics related to the query terms. Since the term clusters cab be used for query expansion, we can get a new document set related to a more specific topic.

There is another way to handle search results. The Findex system [Mak05] does not classify the retrieved documents ; instead it extracts terms or phrases (i.e., a sequence of two or more terms). The extracted terms or phrases represent the features of the documents included in the search results. When users select one of the terms or phrases, the system shows the documents containing it. Findex system is able to support users who input ambiguous queries when the search fails to retrieve relevant documents at the top rank because the results contain various topics. It tries to show terms and phrases corresponding to document categories regardless of topic. In contrast, I want to make individual term clusters, each corresponding to a topic contained in the search results.

Result catgorization and query refinement methods only categorize the search results, and may have difficulty labeling the groups. Findex considers the salient terms to show clusters, and has the merit of showing the search result categories. Besides these methods, faceted search is another approach to organize search results [Hea06a]. Facets denote attributes in various orthogonal sets of categories [Adk05] [Den03] [HSC06] [Mor05]. For example, let's assume that we go to a shopping site for jewelry. In such a situation we can think of "material" and "type" as examples of facets, and silver and gold as examples of values of "material" facet. Although faceted classification was developed long ago by S. R. Ranganathan in the 1930's, faceted search has recently become a topic of renewed interest*. A Faceted search puts a new spin on the idea of a parametric search. A parametric search presents every available field with every available value [Sea06], and those fields and values are fixed regardless of keywords. In contrast, a faceted search complements keyword searches by showing the facets contained in the search results. It is a problem for the faceted search that each document contained in the search results has no tags for facets. I believe that the method described in this thesis provides good candidates to generate facets because it tries to extract terms exclusively related to one of the topics contained in the search results and to make term clusters for each topic.

Keyword Extraction Numerous studies have been done on keyword extraction. Most of them report methods for extracting topic-centric terms, such as technical terms and proper nouns [CG95] [RJ05]. Rennie and Jaakkola [RJ05] introduced a new informativeness measure, the Mixture score, which focuses on the difference in log-likelihood between a mixture model and a simple unigram model, to identify informative words. They compare it against a number of other informativeness criteria, including the Inverse Document Frequency (IDF) and Residual IDF (RIDF) [CG95]. While the results show their measure works well when compared with existing methods, the documents they used are all posts to a restaurant discussion bulletin board, so these results cannot be seen as conclusive.

Sanderson et al. [SC99] [LC00] extracted terms and made concept hierarchies from the search results. They used term co-occurrences to find strong relationship between terms. And also, H. Joho et al. [JCSB02] examines an hierarchical presentation of the expansion terms which are automat-

*Workshop on Faceted Search was held at SIGIR'06 (http://facetedsearch.googlepages.com/).

ically generated from a set of retrieved documents, organisad in a general to apecific manner, and visualised by cascade menus. On the other hand, in this work, I aim to improve the search results with topical term clustering. I first extract topical terms using term co-occurrences and next make clusters with those terms.

Specific terms such as proper nouns and technical terms can represent a specific topic succinctly [OKI04] [OKIY01] [小久 02]. Therefore, I aim to make term clusters with which we can easily understand the topics included in a document set. There have been many studies on well-known term weighting methods [YP97] [Seb02]. I compare those methods with my proposed method in this paper. Other studies have concentrated on term extraction, such as measurement of term representativeness [HNN+00], keyword extraction from one document by using term clustering [MI04] [松尾 02], KeyGraph [ONY98] [大澤 99], and DualNAVI [TNN+] [NIH+99] [TNN+00]. However, the methods proposed in [HNN+00] and in [MI04] do not aim to distinguish different topics included in a set of documents. While KeyGraph and DualNAVI visualize the relationship between terms, I aim to support query refinement by showing term clusters.

There are various methods to look for additional terms used for query expansion, e.g., Robertson's Selection Value (RSV) [Rob90]. However, query expansion methods do not consider that the search results may contain multiple topics. Interactive relevance feedback [BYRN99] [SB90] uses documents selected by users to obtain additional query terms. On the other hand, I think it is easier to browse term clusters divided into topics than to browse many documents showing various topics. Moreover, term clusters can also be used as additional query terms to refine the original query.

And also, there is a technique proposed by Fox et al. called query splitting [FDNY+06] [YDNF05]. Their system helps with "query splitting" finding two subqueries when a user provides a single statement. It then works to find the pathways that connect those subqueries. It can support a closed discovery model, where the two topics to connect are given by the user, or an open discovery model, where the two topics are inferred from the query and the collection content.

2.2 Organizing Search Results

2.2.1 Scatter/Gather

Document clustering has been extensively investigated as a methodology for improving document search and retrieval such as Scatter/Gather [CKPT92][HP96] . The Scatter/Gather system was proposed as a document browsing method to organize the search results in the early 1990's. The system is based on the general assumption that mutually similar documents will tend to be relevant to the same queries. Initially the system *scatters* the collection into a small number of document groups, or clusters, and presents short summaries of them to the user (See Figure 2.1). The user selected one or more of the groups for further study, and the selected groups are *gathered* together to form a subcollection (See Figure 2.1). The system uses seed-based partitional clustering algorithms, that is, Buckshot and Fractionation. The short summaries of the group are generated from the *central words* appearing most frequently in the group as a whole.

2.2.2 Findex

Long web search result lists can be hard to browse, and it is said that a categorizaion algorithm and a filtering interface are usefulness to the searchers. To check the usefulness in real settings, Mika [Mak05] provided a categorizing web search user interface showed in Figure 2.2 to 16 users for a two month period, and the users' interactions with the system were logged. The results show that categories are successfully used as part of users' search habits. The categories are helpful when the result ranking of the search engine fails because the users are able to access results located far in the search result list with the categories. They can also formulate simpler queries and find needed results with the help of the categories.

2.2.3 Faceted Search

Faceted search is the next approach to organize the search results [Hea06a]. Facets denote attributes in various orthogonal sets of categories [Adk05] [Den03] [HSC06] [Mor05]. For example, assume that we use shopping site for jewelry. "Material" and "type" are examples of facets; silver and gold are examples of facet values in the "material" facet. Although faceted classification was already

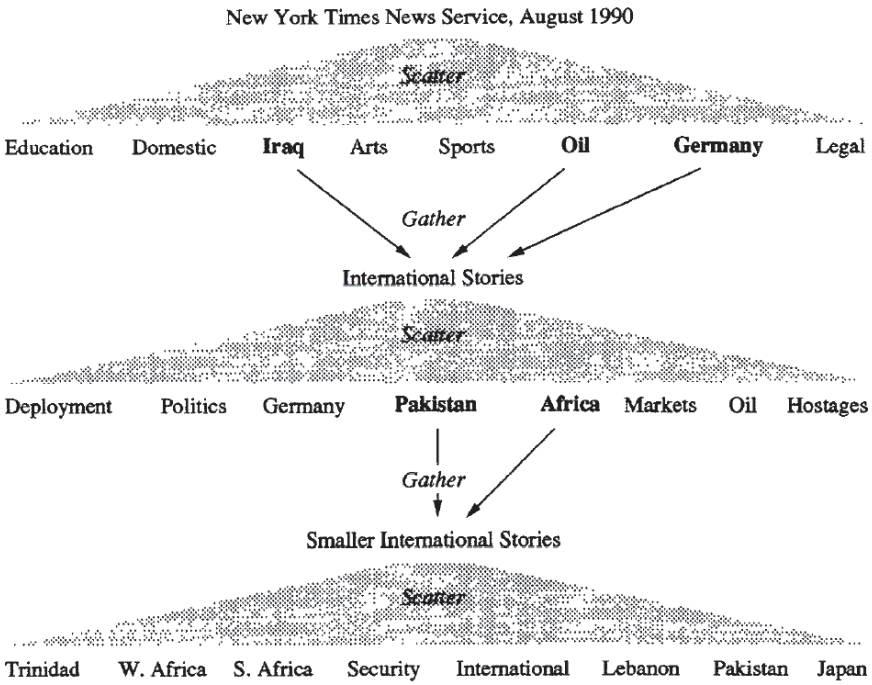


Figure 2.1: Illustration of Scatter/Gatter [CKPT92].

Figure 2.2: Filtering Findex search user interface [Mak05]. Categories on the left, filtered results on the right.

developed by S. R. Ranganathan in 1930's, faceted search is now a hot new topic. Workshop on Faceted Search was held at SIGIR'06 (http://facetedsearch.googlepages.com/).

Faceted search is a new spin on parametric search. Parametric search presents every available field with every available value [Sea06], and those fields and values are fixed regardless of keywords. In contrast, faceted search is a successful complement to keyword searching by showing the facets contained in the search results. It is the problem for the faceted search that each document contained in the search results has no tags for facets. Faceted search aims to combine navigational and direct search to leverage the best of both approaches. Faceted search systems assume that the records are organized into multiple independent facets, rather than a single taxonomy [Tun06].

Flamenco project by Hears et al. developed some systems for database as Faceted Search [Hea06b][YSLH03] [Hea06a]: Faceted Search for Recipes [†] , for Nobel prize winners[‡] , and for architecture slides [§]. These systems are for the database of recipes, Nobel prize winners, and the UC Berkeley Architecture Slide library. Ross et al. also aim to search faceted databases [RJ04].

On the other hand, Wisam et al. aim to automatically identify facets that can be used to browse a collection of free-text documents [DDI06]. To identify the candidate facet terms, they identify terms that were rather infrequent in the original database, but are frequent in the database with expanded documents. Their approach on identifying facet terms is conceptually similar to the skew divergence of Lee [Lee], which is used to identify substitute terms.

2.3 Keyword Extraction

2.3.1 Keyword Spices

The information retrieval on the web has come into wide use in our society, but gathering information on the web is a difficult task for a novice searcher. This is because the searcher must have skill to find the relevant patges from the large number of documents returned, which often cover a wide variety of topics. Kokubo et al. proposed a new method that improves search performance by adding the domain-specific keywords, called *keyword spices*, to the searcher's input query [OKI04][OKIY01][小久 02]. The keyword spice model of building domain-specific web search en-

[†]http://orange.sims.berkeley.edu/cgi-bin/flamenco.cgi/recipes-automated/Flamenco
[‡]http://orange.sims.berkeley.edu/cgi-bin/flamenco.cgi/nobel/Flamenco
[§]http://orange.sims.berkeley.edu/cgi-bin/flamenco.cgi/spiro/Flamenco

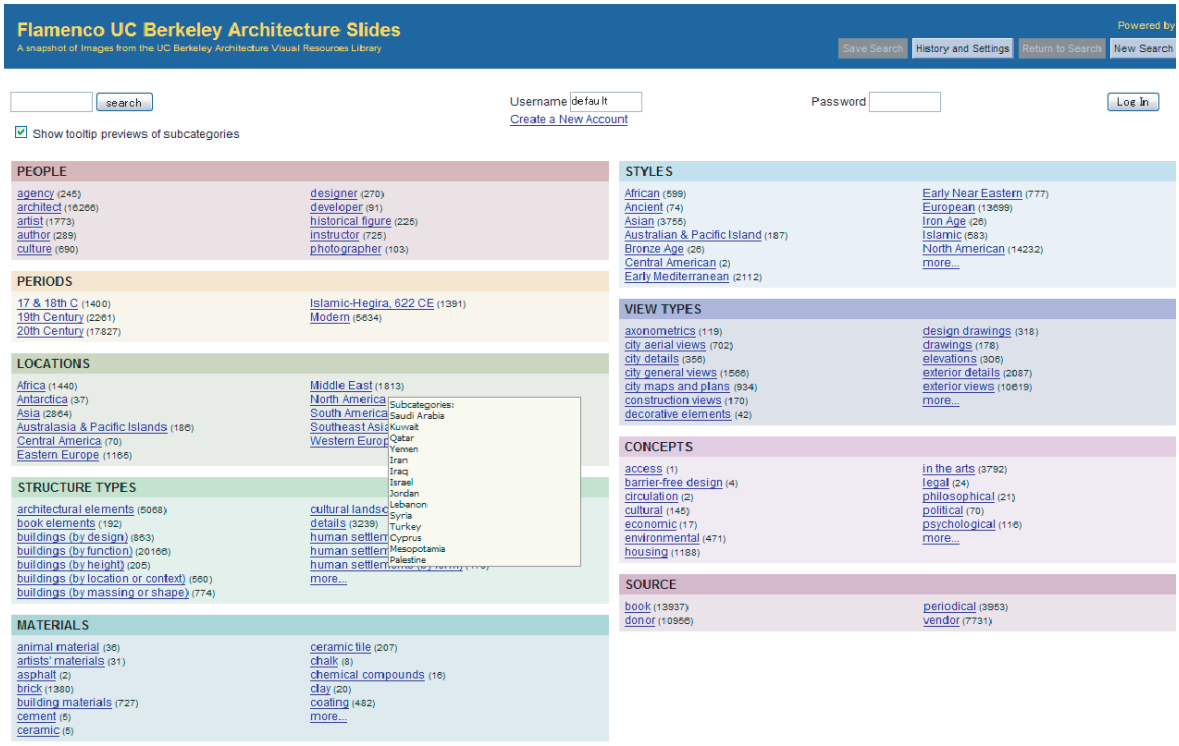


Figure 2.3: Hierarchical facet navigation in Flamenco [Hea06b]. The system uses the database of Flamenco US Berkeley Architecture Slide Library.

gines is the reverse of the filtering model. Suppose, for instance, you want to find recipes on the web. When you input "beef" as a query to a search engine, you will find few recipes, but many other pages on disease, farming, and trading in the top-ranked pages. In contrast, when you input "beef pepper", you will be surprised to find that most of the returned pages are recipes. Kokubo et al. aims to find keyword spices such as "pepper" which is effective to find recipes if it is used as an addicional query term to be added to the original query term such as "beef" when the searcher requires recipes. The keyword spice model does not filter documents returned by a general-purpose search engine. Instead, it extends the searcher's input query with a domain-specific Boolean expression and passes the extended query to a general-purpose search engine. The keyword spice extraction algorithmis is based on decision-tree learning algorithm, and it can extract keyword spices as a disjunctive normal form of keywords from documents on the web.

2.3.2 Representativeness

The method proposed by Hisamitsu et al. [HNN+00] compares the term frequency distributions in an entire document set with those in the set of documents containing a specific term t. When a large discrepancy exists between them, t is said to have *representativeness*. This method estimates a discrepancy similar to ours. However, my concern lies in the *direction* of the discrepancy. We ask whether the frequency of terms other than t in an entire set is higher or lower than that in a set of documents including t. When the latter is less than the former with respect to a large number of terms, we say that t has Tangibility.

2.3.3 Keyword Extraction from a Single document using Word Co-occurrence Statistical Information

Matsuo et al. [MI04] proposed a term extraction method based on term co-occurrences. Their method combines term ranking by the χ^2 measure with term clustering. However, this method is designed for application to a single document. In contrast, I aim to disambiguate a query by finding the terms corresponding to distinct topics latent in a set of hundreds of retrieved documents. Therefore, I have proposed a new measure for term extraction.

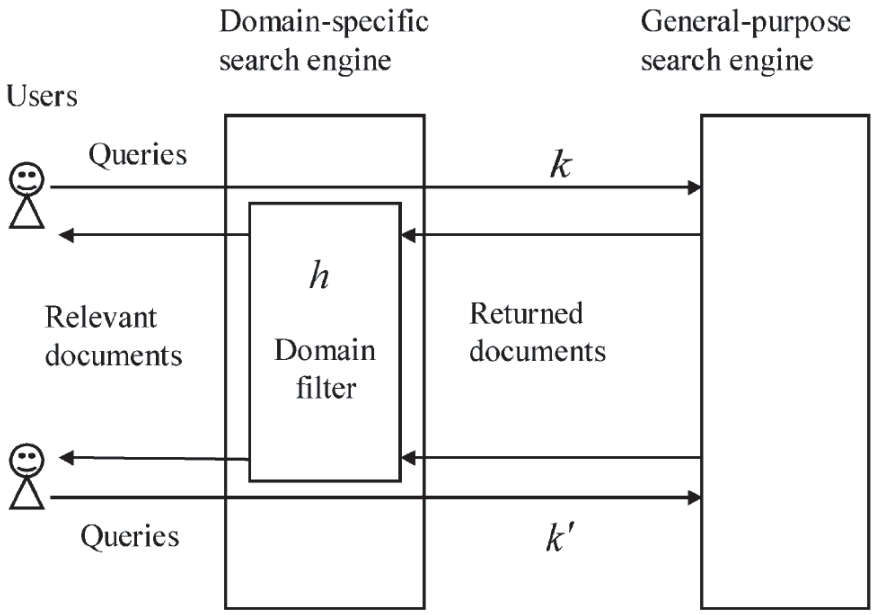


Figure 2.4: Filtering model of building domain-specific web search engines [OKI04].

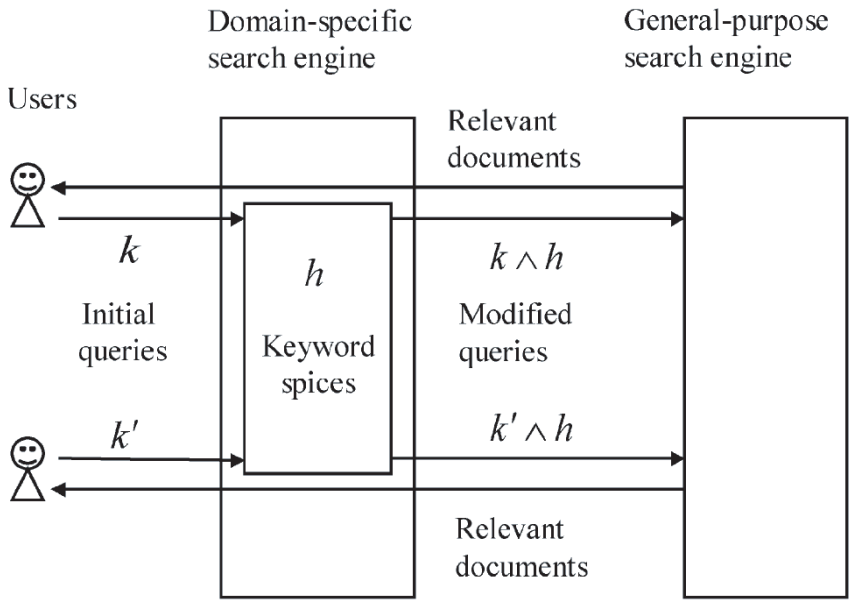


Figure 2.5: Keyword spice model of building domain-specific web search engines [OKI04].

2.4 Recent Search Engines

A number of search engines that organize retrieved results have been developed recently [松岡 05]. Some of them are meta search engines, meaning it combines results from a variety of different sources, e.g., Google, Yahoo, and MSN. Others are document clustering search engines.

For instance, Clusty[¶] is Vivisimo's meta-search engine that offers enhanced features such as clustering. Aside from the usual search tabs, Clusty includes Gossip and Jobs. It also allowed us to narrow our search using quotes and the + sign. Our results show clusters on the left. We can easily see how many items are in each cluster. Clicking a cluster filters the search results on the right side. Clustering is the automatic organization of search results into groups or clusters. It differs from other techniques (classification, taxonomy building, tagging, etc.) in that it requires no pre-processing or human intervention. Cluster labels are intelligently created from the words and phrases contained within the search results.

[¶]http://clusty.com/

Chapter 3

Tangibility: A New Measure for Topical Term Extraction

3.1 Hypothesis of Tangibility

To cope with ambiguities in queries, I propose a new method for selecting terms. The aim of my method is to find more specific terms than the query terms the user has given; these specific terms can match more easily with distinct topics and resolve query ambiguity. Here I introduce a new concept called *Tangibility*. I say that a term has *Tangibility* when it keeps a fairly close relationship with the given query and, at the same time, is strongly related to a distinct topic, regardless of whether or not the topic is principal in the retrieved document set. I measure the Tangibility of a term t by focusing on the variety of terms frequently co-occurring with t. To obtain numerical estimates of Tangibility, I formulate my hypothesis as follows:

> *A term co-occurring frequently with a limited number of terms in a retrieved document set can establish a distinct topic in the document set.*

I call this the *hypothesis of Tangibility*. I say two terms *co-occur* when they appear in the same document. Each term is counted only once, even if it appears many times within the document. In the following subsection, I propose three numerical estimates for term Tangibility: TNG1, TNG2 and TNG.

3.2 Notation and Definitions

In this paper, I say two terms co-occur when they appear in the same document. Let $P(t_i)$ be the occurrence probability of a term t_i. $P(t_i)$ is defined as the number of documents in which t_i appears divided by the total number of documents. Let $P(t_j|t_i)$ be the occurrence probability of t_j among the documents including t_i. $P(t_j|t_i)$ is defined as the number of documents in which t_i and t_j co-occur divided by the number of documents where t_i appears. In the same way, let $P(\neg t_j|t_i)$ be the nonoccurrence probability of t_j among the documents including t_i. Let U be a document set, and let $U(t_i)$ be a document set in which t_i appears. Let S be a document subset, and let $S(t_i)$ be a document subset in which t_i appears. For example, U is a corpus for retrieval, and S is a set of retrieved documents.

3.3 Formulations of Tangibility

3.3.1 TNG1: First Formulation of Tangibility

My first numerical estimate of Tangibility, denoted by TNG1, is based on the hypothesis described in Section 3.1. To obtain TNG1, I introduce the average number of terms that appear in documents that include term t_i, and denote it by $F(t_i)$. More formally, $F(t_i)$ is defined as follows:

$$F(t_i) = \frac{\sum_{d \in S(t_i)} (V(d) - 1)}{|S(t_i)|}. \tag{3.1}$$

Here, $V(d)$ denotes the number of terms in document d. $F(t_i)$ shows how many terms appear with t_i in S. Therefore, we can regard a term with small $F(t_i)$ as a term representing a distinct topic. However, a term having small $|S(t_i)|$ intrinsically has small $F(t_i)$. I therefore introduce an additional component into my formula so that terms of small $|S(t_i)|$ should not always be regarded as having Tangibility. Consequently, I obtain the following formula as TNG1, which expresses the Tangibility of a term t_i [HWA06][若木 05]*:

$$TNG1(t_i) = \frac{|S(t_i)|^2}{|U(t_i)|} \cdot \frac{1}{F(t_i)}, \tag{3.2}$$

*TNG1 is called as AR1 in [若木 05].

where the first half is obtained by multiplying $|S(t_i)|$ by $|S(t_i)|/|U(t_i)|$. $|S(t_i)|$ is simply the document frequency of t_i in S and indicates how strongly t_i is *unconditionally* related to S. In contrast, $|S(t_i)|/|U(t_i)|$ indicates how strongly t_i is related to S *in comparison with* U.

3.3.2 TNG2: Second Formulation of Tangibility

To provide the second formulation TNG2, I rewrite Equation (3.1) as follows:

$$F(t_i) = \sum_{t_j \neq t_i} \frac{|S(t_i \wedge t_j)|}{|S(t_i)|}, \tag{3.3}$$

where $|S(t_i \wedge t_j)|$ is defined to $|S(t_i) \cap S(t_j)|$. Since I can interpret $|S(t_i \wedge t_j)|/|S(t_i)|$ as the probability of the occurrence of t_j among the documents including t_i, we denote it by $P(t_j|t_i)$. According to TNG1, t_i has Tangibility when $\sum_{t_j \neq t_i} P(t_j|t_i)$ is small. In contrast, I devise the second formulation, TNG2, by requiring $P(t_j|t_i)$ to be *smaller than* $P(t_j)$ for a large number of t_js ($j \neq i$). TNG1 and TNG2 share the same intuition. However, I introduce an elaboration into TNG2, i.e., the comparison of $P(t_j|t_i)$ with $P(t_j)$. For the discrepancy evaluation of the two probability distributions, Kullback–Leibler Divergence (KLD) is often used. In my case, $P(t_j|t_i)$ and $P(t_j)$ are to be compared. Moreover, the event complementary to the occurrence of t_j is the non-occurrence of t_j, denoted by $\neg t_j$. Therefore, the KLD for my evaluation can be written as:

$$KL(t_j; t_i) = P(t_j|t_i) \log \frac{P(t_j|t_i)}{P(t_j)} + P(\neg t_j|t_i) \log \frac{P(\neg t_j|t_i)}{P(\neg t_j)}.$$

However, $\sum_{t_j \neq t_i} KL(t_j; t_i)$ cannot evaluate the Tangibility of t_i, because this sum is large when any of the following two conditions holds for many t_js:

$$\text{(a) } P(t_j|t_i) \log \frac{P(t_j|t_i)}{P(t_j)} < 0 \ \left(\text{and thus } P(\neg t_j|t_i) \log \frac{P(\neg t_j|t_i)}{P(\neg t_j)} > 0 \text{ also holds}\right)$$

$$\text{(b) } P(t_j|t_i) \log \frac{P(t_j|t_i)}{P(t_j)} > 0 \ \left(\text{and thus } P(\neg t_j|t_i) \log \frac{P(\neg t_j|t_i)}{P(\neg t_j)} < 0 \text{ also holds}\right)$$

Only (a) is important for the Tangibility of t_i. Therefore, I propose a new measure, called Signed Kullback–Leibler (SKL), as follows:

$$SKL(t_j;t_i) = -P(t_j|t_i)\log\frac{P(t_j|t_i)}{P(t_j)} + P(\neg t_j|t_i)\log\frac{P(\neg t_j|t_i)}{P(\neg t_j)}. \tag{3.4}$$

SKL is derived from the KLD by changing the sign of the first term. Consequently, I propose the following as the second formula for Tangibility [HWA06][若木 05]†:

$$TNG2(t_i) = \frac{|S(t_i)|^2}{|U(t_i)|}\cdot\sum_{t_j\neq t_i} SKL(t_j;t_i).$$

3.3.3 TNG: Improved Formulation of Tangibility

1. **Formulation of TNG**

   I will introduce a formula called TNG to calculate how high a term's tangibility is. First, the following equation measures how much the probability of t_j's appearance increases by adding the condition that t_i appears.

$$\Delta_{t_i}(t_j) \quad = \quad P(t_j|t_i)\times\log\frac{P(t_j|t_i)}{P(t_j)} \tag{3.5}$$

   By setting $F_i = \{t_j|\Delta_{t_i}(t_j) > 0\}$, I obtain the formula for TNG, as follows [若木 06b][若木 06a]‡.

$$TNG(t_i) = \frac{\sum_{t_k\in F_i}(\Delta_{t_i}(t_k))}{|F_i|} \tag{3.6}$$

   Eq. (3.6) weights terms, and I can rank terms by using those weights. If a term has low frequency, I must avoid the problem of data sparseness. Therefore, I use Dirichlet smoothing[HLF05], as follows.

$$P(t_j|t_i) = \frac{|S(t_i)\cap S(t_j)| + \alpha|S(t_j)|}{|S(t_i)| + \alpha|S|} \tag{3.7}$$

2. **The Meaning of TNG**

   Eq. (3.5) is part of the Kullback–Leibler Divergence (KLD), which is defined as follows.

†TNG2 is called as AR2 in [若木 05].
‡TNG is called as TNG3 in [若木 06a].

$$KLD(t_j;t_i) = \quad P(t_j|t_i)\log\frac{P(t_j|t_i)}{P(t_j)} + P(\neg t_j|t_i)\log\frac{P(\neg t_j|t_i)}{P(\neg t_j)} \tag{3.8}$$

The second half of Eq.(4.3) is positive when t_j's occurrence probability is decreased by adding the condition that t_i occurs. This case does not correspond to the hypothesis of tangibility, so I only use the first half of Eq.(4.3). Eq.(3.5) is positive if and only if that probability increases. Moreover, Eq.(3.5) is used in Eq.(3.6) only when $\Delta_{t_i}(t_j) > 0$. This means Eq.(3.6) calculates the average of $\Delta_{t_i}(t_j)$ only when t_j satisfies $\Delta_{t_i}(t_j) > 0$. If Eq.(3.6) calculates the average for all terms co-occurring with t_i, an inconvenience arises: we cannot distinguish between the cases in which many terms co-occur with t_i less frequently and some specific terms co-occur with t_i frequently. Thus, Eq.(3.6) is certainly formulated based on the concept of tangibility. Eq.(3.6) is a revised version of my previous formulations of TNG1 and TNG2.

3.4 Comparison Methods

Many recent studies have proposed various term-weighting methods for term extraction. Some of them use term co-occurrence frequencies as in my formulations of Tangibility.

3.4.1 Mutual Information (MI)

$$\begin{aligned} MI(t_j;t_i) \quad &= \quad P(t_i)\{P(t_j|t_i)\log\frac{P(t_j|t_i)}{P(t_j)} + P(\neg t_j|t_i)\log\frac{P(\neg t_j|t_i)}{P(\neg t_j)}\} \\ &+ \qquad P(\neg t_i)\{P(t_j|\neg t_i)\log\frac{P(t_j|\neg t_i)}{P(t_j)} + P(\neg t_j|\neg t_i)\log\frac{P(\neg t_j|\neg t_i)}{P(\neg t_j)}\} \end{aligned} \tag{3.9}$$

3.4.2 Kullback-Leibler Divergence (KLD)

$$KLD(t_j;t_i) \quad = \quad P(t_j|t_i)\log\frac{P(t_j|t_i)}{P(t_j)} + P(\neg t_j|t_i)\log\frac{P(\neg t_j|t_i)}{P(\neg t_j)}. \tag{3.10}$$

3.4.3 Chi-square

$$\begin{aligned}\chi^2(t_j;t_i) &= \frac{\{P(t_j|t_i)-P(t_j)\}^2}{P(t_j)}+\frac{\{P(\neg t_j|t_i)-P(\neg t_j)\}^2}{P(\neg t_j)} \\ &+ \frac{\{P(t_j|\neg t_i)-P(t_j)\}^2}{P(t_j)}+\frac{\{P(\neg t_j|\neg t_i)-P(\neg t_j)\}^2}{P(\neg t_j)}\end{aligned} \tag{3.11}$$

3.4.4 Robertson's Selection Value (RSV)

RSV[Rob90] is a term selection method for query expansion. This method can measure the relevance ratio of each term to a document set selected for a purpose such as information retrieval. Note that RSV does not use term co-occurrence probabilities. $RSV(t_i)$ is defined as follows.

$$RSV(t_i) = (\frac{|S(t_i)|}{|S|}-\frac{|U(t_i)|}{|U|})\times\left\{k\times\log\frac{|U|}{|U(t_i)|}+(1-k)\times\log K\right\} \tag{3.12}$$

Here, $K=\frac{|S(t_i)|+0.5}{|S|-|S(t_i)|+0.5}/\frac{|U(t_i)|-|S(t_i)|+0.5}{|U|-|U(t_i)|-|S|+|S(t_i)|+0.5}$.

k is a parameter, and I set $k=0.5$ after appropriate tuning.

3.5 Term Clustering using Extracted Terms

3.5.1 Similarity between Terms

It is important to define an appropriate similarity between terms for term clustering. In this paper, let the similarity between term t_i and term t_j be as follows.

$$Sim(t_i,t_j)=\frac{|S(t_i)\cap S(t_j)|}{|S(t_i)\cup S(t_j)|} \tag{3.13}$$

I set $Sim(t_i,t_j)=0$ if $|S(t_i)\cap S(t_j)|<5$ in my experiment.

3.5.2 Similarity between Clusters

The distance between clusters is defined as follows[DHZ+01][DH02].

$$Sim(C_1,C_2)=\frac{s(C_1,C_2)}{(s(C_1,C_1)+|C_1|)\times(s(C_2,C_2)+|C_2|)} \tag{3.14}$$

Here, I defined $s(C_1, C_2)$ as follows.

$$s(C_1, C_2) = \sum_{t_i \in C_1} \sum_{t_j \in C_2} Sim(t_i, t_j) \tag{3.15}$$

I set $M = 10$ and use the top-ranked 100 terms. Where the distance between any two clusters is infinite, no clusters are merged, because an infinite distance indicates that no pairs of terms from different clusters co-occur in the same documents.

3.5.3 Distributional Clustering Algorithm

I use distributional clustering [BM98][ISDK03], proposed by Baker et al. This clustering algorithm uses the ranks of terms when it makes clusters, so the generated clusters are different for different methods of ranking terms.

Distributional Clustering

1. Sort the entire vocabulary by MI with the class variable.
2. Initialize M singleton clusters with the top M words.
3. Compute the intercluster distances between every pair of clusters.
4. Loop until all words have been put into one of the M clusters.

i) Merge the two clusters that are most similar resulting in $M-1$ clusters.

ii) Add a new singleton cluster consisting of the next word from the sorted list of words.

iii) Update the intercluster distances.

Baker et al. ranked the terms by using mutual information (MI). However, in this paper, I exchange MI for other term weighting methods such as TNG.

Chapter 4

Experiment for Tangibility

4.1 Experiment for Query Refinement with TNG1 and TNG2

4.1.1 Overview

This work explores techniques that discover terms to replace given query terms from a selected subset of documents. The Internet and digital libraries have allowed access to large numbers of documents archived in digital format. However, most users are not experts in every field and cannot formulate queries that narrow the search to the context they have in mind. Accordingly, I propose a method for extracting terms from searched documents to replace user-provided query terms as semiautomatic query expansion.

My method analyzes the co-occurrence of terms in the top-ranked documents of the initial search result and extracts terms that are important in terms of their *tangibility*. When a term refers to a specific concept or denotes a particular thing, I say the term has *tangibility*. A proper noun is a typical example of a term having *tangibility*. My method is based on the following observation: we can easily disambiguate a short query by adding just one term that has *tangibility*(See Section 3.1). Additionally, I proposed two formulae for Tangibility as I mentioned in the Section 3.3.1 and 3.3.2, that is TNG1 and TNG2. My method works regardless of the retrieval method I use. Moreover, my method can extract expansion terms without using additional data such as word networks or structural directories of concepts.

In this section, I will discribe my experiment for those formulae. The results show that my

method is successful in discovering terms that can be used to narrow the search. Moreover, we acquire new knowledge specialized in the context of the query if we did not know one or more of the terms presented by my method.

4.1.2 Arrangement of Equations to be Compared

Before describing my experiment in detail, I present the various term weighting schemes that I tested. Term weight $W(t_i)$ for term t_i is computed by multiplying two weights:

$$W(t_i) = \frac{N_S(t_i)^2}{N_U(t_i)} \times \{CW(t_i)\}^{\sigma}.$$

The former weight was introduced in Section 2.2. It represents the importance of t_i in the retrieved document set S, and depends only on the occurrence frequency of t_i. I obtain the latter weight $CW(t_i)$ by summing $cw(t_i, t_j)$ for all t_j not equal to t_i as follows:

$$CW(t_i) = \sum_{t_j \neq t_i} cw(t_i, t_j).$$

$cw(t_i, t_j)$ is computed based on the co-occurrence of t_i and t_j with respect to the particular property of terms I intend to evaluate. σ takes a value of 1 or –1. When I want an increase in $CW(t_i)$ to contribute to an increase of $W(t_i)$, σ is set to 1. On the other hand, when I want a decrease in $CW(t_i)$ to contribute to an increase of $W(t_i)$, σ is set to –1.

Many recent studies have proposed various term weighting methods for term extraction. Some of them use term occurrence frequencies as in my formulations of *tangibility*. Therefore, I compared eight term weighting methods[YP97]: TNG1, TNG2, UnitWeight, CF, Mutual Information (MI)[YH04], Kullback–Leibler Divergence (KLD), χ^2 test, and Robertson's Selection Value (RSV)[Rob90]. The detail of each formulation appears in Table 4.1.

The term weighting method *UnitWeight* is so called because $CW(t_i) = 1$. This method ignores the effect of term co-occurrence. That is, *UnitWeight* is intended to reveal how the difference of $CW(t_i)$ works in each of the other term weighting methods.

CF (for Co-occurrence Frequency) is prepared for ranking terms based on the assumption of term importance, contrary to TNG1. According to this assumption, I give larger weights to terms that frequently co-occur with many other terms.

MI, KLD, and χ^2 use $cw(t_i, t_j)$ to measure the discrepancy between $P(t_j|t_i)$ and $P(t_j)$. Therefore, they are all based on nearly the same intuition about term importance. With these measures, however, I cannot distinguish whether

$P(t_j|t_i) > P(t_j)$ or $P(t_j|t_i) < P(t_j)$.

4.1.3 Experimental Procedure

I used a document set prepared for the NTCIR3 Web Retrieval Task[EOI+03]. This set includes about ten million Web pages written in Japanese. we denote this Web page set by U. The Web pages in U are decomposed into terms by using a morphological analyzer MeCab[MeC] equipped with a Japanese dictionary ipadic-2.5.1[ipa]. There are 47 queries prepared for the NTCIR3 Web task, and each query includes two or three query terms. First, I issued the queries and obtained the top 1000 Web pages for each query (see Step 2 in Fig. 4.1). Although my experiment adopted an Okapi-type term-weighting scheme for Web page retrieval[FTZ], my method can be applied to the search results obtained with other term-weighting schemes. From the top 1000 pages of each of the 47 retrieval results, I gathered terms appearing in five or more pages. I obtained about 10,000 terms for each query. I did not delete stop words. Next, I computed the eight term weights described in 4.1(see Step 3 in Fig. 4.1). As a result, I obtained eight term rankings by sorting the terms with respect to their eight kinds of weights. For every term ranking, I added each of the top five terms (a, b, c, d, and e) separately to the original query term set $\{A, B, C\}$ and made five expanded sets of query terms $\{A, B, C, a\}$, $\{A, B, C, b\}$, ..., $\{A, B, C, e\}$. Finally, I retrieved the Web pages with these expanded query term sets. Consequently, I obtained five search results for each query. I computed the average precisions of these five results by using `trec_eval`[TRE](see Step 4 in Fig. 4.1). Of these five average precisions, I kept only the best one, because this average precision can be taken as the performance measure of the information retrieval most desirable for users who are supposed to issue the corresponding query. Finally, I regarded the mean of the best average precisions of the 47 queries as the overall precision for each term-weighting method (see Step 5 in Fig. 4.1).

Table 4.1: Formulations used in my experiments. In this table, I replace some expressions with symbols as follows: $P(t_j|t_i) = A$, $P(t_j|\neg t_i) = B$, $P(\neg t_j|t_i) = C$, $P(\neg t_j|\neg t_i) = D$, $\frac{N_S(t_i)^2}{N_U(t_i)} = U$.

method	$cw(t_i, t_j)$	$W(t_i) = U \cdot \left\{\sum_{t_j \neq t_i} cw(t_i, t_j)\right\}^{\alpha}$
TNG1	$\frac{N_S(t_i \cap t_j)}{N_S(t_i)}$	$U \cdot \left\{\sum_{t_j \neq t_i} cw(t_i, t_j)\right\}^{-1}$
TNG2	$-A \log \frac{A}{P(t_j)} + C \log \frac{C}{P(\neg t_j)}$	$U \cdot \sum_{t_j \neq t_i} cw(t_i, t_j)$
UnitWeight	-	$U \times 1$
CF	$\frac{N_S(t_i \cap t_j)}{N_S(t_i)}$	$U \cdot \sum_{t_j \neq t_i} cw(t_i, t_j)$
MI	$P(t_i)\left\{A \log \frac{A}{P(t_j)} + C \log \frac{C}{P(\neg t_j)}\right\}$ $+P(\neg t_i)\left\{B \log \frac{B}{P(t_j)} + D \log \frac{D}{P(\neg t_j)}\right\}$	$U \cdot \sum_{t_j \neq t_i} cw(t_i, t_j)$
KLD	$A \log \frac{A}{P(t_j)} + C \log \frac{C}{P(\neg t_j)}$	$U \cdot \sum_{t_j \neq t_i} cw(t_i, t_j)$
χ^2 test	$\frac{\{A - P(t_j)\}^2}{P(t_j)} + \frac{\{C - P(\neg t_j)\}^2}{P(\neg t_j)}$ $+\frac{\{B - P(t_j)\}^2}{P(t_j)} + \frac{\{D - P(\neg t_j)\}^2}{P(\neg t_j)}$	$U \cdot \sum_{t_j \neq t_i} cw(t_i, t_j)$

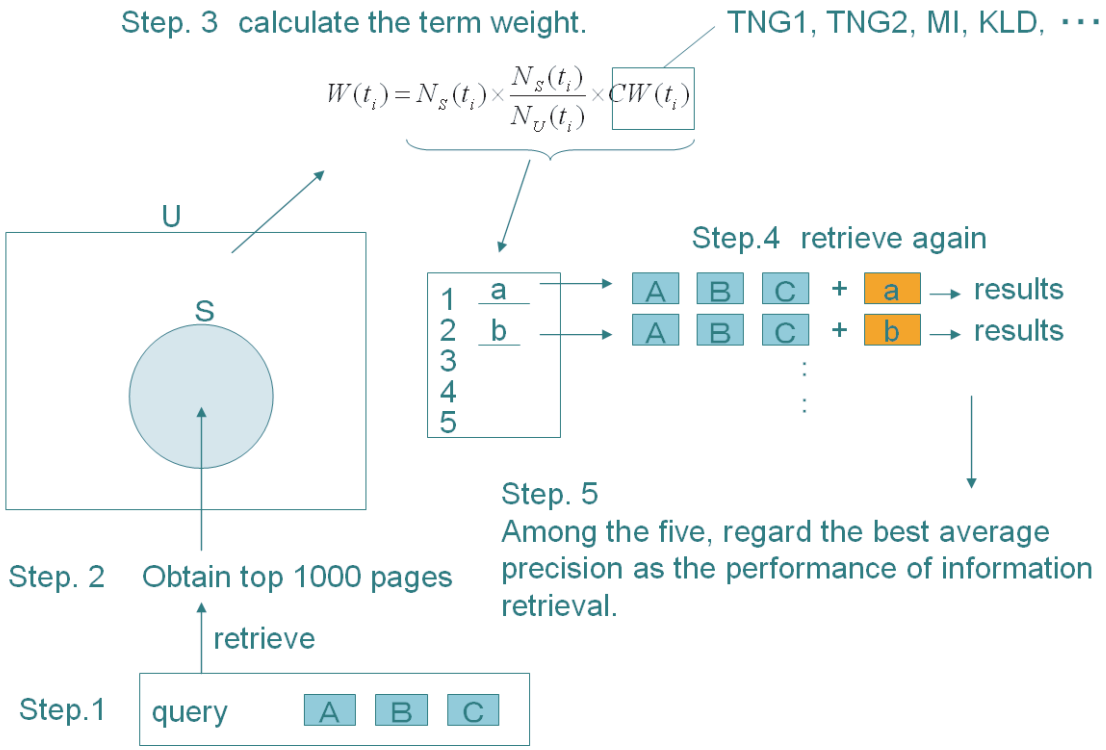


Figure 4.1: Experiment procedures.

4.1.4 Experimental Results

For the original 47 queries, I obtained 0.1606 as the overall precision and regarded it as the baseline. Among the eight term-weighting formulae, TNG1, TNG2, and RSV significantly increased overall precision (Table 4.2). TNG2 achieved the best overall average precision with an 18.2% improvement. The most important point is that TNG1 and TNG2 showed qualitative differences from RSV. While RSV tends to extract terms having general meanings, TNG1 and TNG2 can extract many technical terms used in specific domains relative to the query. For a query including "loudspeaker", "comparison", and "evaluation", my method extracted such technical terms as "woofer" and "bass reflex" (Table 4.3). For a query involving "the World Tree", "Norse mythology", and "name", my method extracted "Yggdrasill", which is the name of a mythological tree in Norse mythology and is a synonym of "the World Tree" (Table 4.3).

4.1.5 Summary

I proposed a co-occurrence-based measure, called Tangibility, for term extraction to disambiguate queries. My experiments obtained very interesting results worthy of further investigation. Both of my numerical estimates for term tangibility, TNG1 and TNG2, realized good average precisions. In addition, many of the extracted terms were related to more specific topics than that implied by the original ambiguous query terms. My method may be used as a key component of a system that helps users to discover specific topics from a given corpus simply by using fairly general terms as search keywords. As future work, I plan to propose a method of clustering the terms that have Tangibility; I will test to determine whether the term clusters correspond to distinct topics implied by the initial query terms.

4.2 Comparative Experiment on Extracting Topical Terms

4.2.1 Overview

I propose a method for supporting query refinement by using clusters of topical terms extracted from a retrieved set of documents. I assume that a topic is implied by a specific set of terms that frequently co-occur in the same documents. Therefore, I introduced a new measure of term importance called *tangibility*. A term is said to have tangibility when it frequently co-occurs exclusively with a specific set of terms (See Section 3.1). Additionally, I proposed a formula for tangibility (TNG in Section 3.3.3). Existing methods for term extraction try to extract terms corresponding to the dominant topic of a given document set. However, because such terms are likely to co-occur with a wide variety of other terms, I cannot distinguish isolated topics. In contrast, my method tries to extract terms exclusively related to one of those topics. I divide the extracted terms into clusters by using a distributional clustering algorithm, which leads to agglomerates of terms frequently co-occurring with each other. The ability to show term clusters that individually correspond to one topic contributes to the discovery of good terms for query expansion. When some of the terms contained in the clusters are unfamiliar to the user, they can be used for learning support.

As far as I know, there are no methods for measuring how much a term is related to one topic. Therefore, I propose two new tools; "*Topic Label*" and "*Topical Skewness*". Topic Label indicates the topic to which a given term is most closely related. Topical Skewness shows the degree to which a given term relates exclusively to one topic. I conducted subjective evaluation experiments to determine the suitability of these two tools. Therefore, I used them to compare the qualities of terms extracted by TNG and other methods.

4.2.2 Applying the Proposed Method

I calculated TNG scores (see Section 3.3.3). I used the top 500 terms sorted by document frequency. Next, I used the top 100 terms sorted by TNG for term clustering(See Section 4.3.1). Note that I say two terms co-occur when they both appear in the same document. I set the number of clusters M to 10.

4.2.3 Compared Term Extraction Methods

Many recent studies propose various term-weighting methods for term extraction. Some of them use term co-occurrence frequencies, as does my formulation of Tangibility. I compared TNG with four other term weighting methods: MI [YP97][YH04], KLD, χ-square [Seb02], and RSV [Rob90]. MI, KLD, and χ-square use term co-occurrence, as does TNG, and these methods can measure how t_j's occurrence probability changes by adding the condition that t_i occurs. The weight of t_i for these methods is calculated as follows.

$$W(t_i) = \sum_j X(t_j; t_i) \tag{4.1}$$

Here, $X(t_j; t_i)$ is replaced by $MI(t_j; t_i)$, $KLD(t_j; t_i)$, and $\chi^2(t_j; t_i)$, respectively (See Equation (4.2), (4.3), (4.4)). I use the same smoothing for these methods as I use for TNG. Moreover, RSV uses the whole document set U, e.g., the corpus for retrieval, as well as a document subset S, e.g., search results (See Equation (4.5)).

- **Mutual Information (MI)**

$$\begin{aligned} MI(t_j; t_i) = {} & P(t_i)\{P(t_j|t_i)\log\frac{P(t_j|t_i)}{P(t_j)} + P(\neg t_j|t_i)\log\frac{P(\neg t_j|t_i)}{P(\neg t_j)}\} \\ + {} & P(\neg t_i)\{P(t_j|\neg t_i)\log\frac{P(t_j|\neg t_i)}{P(t_j)} + P(\neg t_j|\neg t_i)\log\frac{P(\neg t_j|\neg t_i)}{P(\neg t_j)}\} \end{aligned} \tag{4.2}$$

- **Kullback-Leibler Divergence (KLD)**

$$KLD(t_j; t_i) = P(t_j|t_i)\log\frac{P(t_j|t_i)}{P(t_j)} + P(\neg t_j|t_i)\log\frac{P(\neg t_j|t_i)}{P(\neg t_j)}. \tag{4.3}$$

- **Chi-square**

$$\chi^2(t_j;t_i) = \frac{\{P(t_j|t_i)-P(t_j)\}^2}{P(t_j)} + \frac{\{P(\neg t_j|t_i)-P(\neg t_j)\}^2}{P(\neg t_j)} + \frac{\{P(t_j|\neg t_i)-P(t_j)\}^2}{P(t_j)} + \frac{\{P(\neg t_j|\neg t_i)-P(\neg t_j)\}^2}{P(\neg t_j)} \quad (4.4)$$

- **Robertson's Selection Value (RSV)**

  RSV [Rob90] is a term selection method for query expansion. This method can measure the relevance ratio of each term to a document set selected for a purpose such as information retrieval. Note that RSV does not use term co-occurrence probabilities. $RSV(t_i)$ is defined as follows.

$$RSV(t_i) = (\frac{|S(t_i)|}{|S|} - \frac{|U(t_i)|}{|U|}) \times \left\{ k \times \log \frac{|U|}{|U(t_i)|} + (1-k) \times \log K \right\} \quad (4.5)$$

  Here, $K = \frac{|S(t_i)|+0.5}{|S|-|S(t_i)|+0.5} / \frac{|U(t_i)|-|S(t_i)|+0.5}{|U|-|U(t_i)|-|S|+|S(t_i)|+0.5}$.

  k is a parameter, and I set $k = 0.5$ after appropriate tuning.

4.2.4 Data Set of the Experiments

Data Types

The experiments required a document set that includes multiple topics and in which each document has labels indicating a topic. A data set for document classification surely has these labels. Moreover, a test collection for the evaluation of information retrieval is accompanied with relevant document sets for each test query, so I can generate a mixture data set containing multiple topics that are indicated by each relevant document set. Therefore, I used both types of data. Furthermore, I used data sets in Japanese and English to ensure the reliability of the experiment in different languages.

When I use a document set for document classification or in web directories, each category indicates a topic. In my experiments, I used a heterogeneous data set that was a mixture of the three largest categories as a pseudo-data set containing multiple topics. If a term weighting method can extract terms each of which is strongly related to one of those topics, the method can be said to work

well. Note, however, that only RSV is a term weighting method for query expansion. It requires a whole document collection denoted by U besides a document subset denoted by S (such as retrieval results). Therefore, all documents in all categories are used as a whole document collection U, and all documents in the three categories are used as retrieval results S.

When I use a document set for information retrieval, each relevant document set for a query indicates a topic. Moreover, RSV uses a whole document collection prepared for the task of information retrieval as U in Equation 4.5.

Data Sets

I used the seven data sets described in Table 4.9. Each data set has categories, and three of the largest categories are mixed to form pseudo-data including multiple topics. The names of the categories are listed in Table 4.10.

- **NTCIR3，NTCIR4** [EOI+03][EOAI04]

  I used relevant documents prepared for the Web tasks in NTCIR3 and NTCIR4 [EOI+03].

- **Dmoz**[*]

  I used documents in the Science directory in the Dmoz web directory[†].

- **Reuters**

  I used Reuters-21578, which was prepared for document classification.

- **Sankei Sports News**[‡]

  I also used back issues of Sankei Sports News on the web. For RSV, I used the NW100G-01 corpus [EOI+03] prepared for NTCIR3 and NTCIR4 as the whole document set U in Equation (4.5) [§].

- **20 Newsgroups**[¶]

[*] http://dmoz.org/
[†] http://dmoz.org/
[‡] http://www.sanspo.com/
[§] In Equation (4.5), I assume that $|S|$ is 3519, which is the number of all documents included in Sankei Sports News, and $|U|$ is $10,253,810 + 3519$, which is the sum of the number of documents included in NW100G-01 and the number of documents included in Sankei Sports News
[¶] http://people.csail.mit.edu/jrennie/20Newsgroups/

I also used 20 Newsgroups prepared for document classification in which each category contains about 1000 documents.

- **NTCIR-CLIR**[CCK+03]

  Finally, I used relevant documents prepared for the tasks of NTCIR-CLIR, which were cross-lingual information retrievals in NTCIR3. These documents were provided by Mainichi Daily News in English.

Properties and Validity of the Data Sets

This experiment used pseudo-data sets including multiple topics. These data sets were made by mixing three document sets, each of which including documents manually classified in the same category. Therefore, the data sets were different from retrieval results in an actual situation. It is possible that actual retrieval results contain several topics at different levels of granularity, and it is difficult to determine such topics properly. Hence, such retrieval results are not suitable for objective and quantitative evaluation. Additionally, as far as I know, there are no data collections that would be useful for evaluating disambiguations of retrieval results. Consequently, I used pseudo-data sets generated by mixing document categories that seem to be clearly divided into topics.

Though the pseudo-data sets can be clearly divided into topics, they contained ambiguities which would be found in actual retrieval results because they were created by mixing documents in the same topic at a higher level of granularity. For example, the data set of Sankei Sports News includes not only articles on soccer, but also articles from two similar categories, Japanese baseball and MLB. The articles from these two categories fall into the same category at a higher level of topic granularity. Therefore, I can say that this data set is valid as a surrogate of actual search results. Also in the data set of NTCIR3, both the documents from the category of Article 9 and those from the category of copyright are related to legal issues. Furthermore, in the data set of Dmoz, all documents come from the directories under the same upper directory, i.e., the Science directory. All documents of the data set of 20 Newsgroups are related to politics, as mentioned before.

Data Preprocessing

The data in Japanese were analyzed by using the MeCab morphological analyzer[‖] with the ipadic-2.5.1 dictionary[**]. The data in English were stemmed by Porter's stemming algorithm, and stop words were eliminated. I used 1000 of the highest document frequency terms in every data set.

4.2.5 Term Evaluation Tools: Topic Label and Topical Skewness

There are various data sets with the correct labels assigned to each document. Therefore, I can use such data sets for performance evaluations when I conduct document clustering or document classification experiments. On the other hand, my method tries to generate term clusters, each corresponding to a topic contained in a given document set. Therefore, I need to test whether a term is strongly related to one of those topics. However, as far as I know, there are no data sets with correct labels assigned to each term. Additionally, it is time-consuming to manually assign correct labels to all terms appearing in the data set. Therefore, to evaluate objectively and numerically whether a term-weighting method can extract topical terms, I developed two new tools: Topic Label and Topical Skewness. Topic Label indicates the topic to which a given term is most closely related. Topical Skewness shows how exclusively a given term relates to one topic. When the data set has document categories, I can assign a Topic Label to every term and compute Topical Skewness for every term. This allows us to use the document category names as the labels assigned as Topic Labels.

The definition of Topical Skewness of a term t_i is as follows. Let $p_j(t_i)$ $(j = 1,2,3)$ be the ratio of the number of documents including t_i within the category $j(j = 1,2,3)$ to the number of documents including t_i. Let $q_j(j = 1,2,3)$ be the ratio of the number of documents contained within the category $j(j = 1,2,3)$ to the total number of documents. Topical Skewness $TS(t_i)$ is defined as:

$$TS(t_i) = \frac{DF(t_i)}{N} \times K(t_i), \tag{4.6}$$

[‖]http://mecab.sourceforge.jp/
[**]http://chasen.naist.jp/stable/ipadic/

where

$$K(t_i) = \sum_j p_j(t_i) \log \frac{p_j(t_i)}{q_j}; \tag{4.7}$$

i.e., $K(t_i)$ denotes the Kullback–Leibler Divergence of $p_j(t_i)$ and q_j.

That is, the more the probability of the each term occurrence ($p_1(t_i)$, $p_2(t_i)$, $p_3(t_i)$) is skewed against one of the labels compared with the probability of the labels attached to the documents containing t_i, the more likely the value of Equation (4.7) will increase. Note that $K(t_i)$ is multiplied by $\frac{DF(t_i)}{N}$ for the purpose of correction. This is because an infrequently occurring term is likely to be skewed against one of the topics even though the term is not strongly related to the topic.

I predict one of the topics most strongly related to t_i by using the following Topic Label equation.

$$TL(t_i) = \arg\max_j p_j(t_i) \log \frac{p_j(t_i)}{q_j} \tag{4.8}$$

That is, $TL(t_i) = k$ when $p_k(t_i) \log \frac{p_k(t_i)}{q_k}$ becomes maximum. $TL(t_i)$ is considered as the label to be attached to t_i. Moreover, I consider $TS(t_i)$ to be the Topical Skewness against the category k.

4.2.6 Human Subjective Evaluation

Participants and Procedure

I conducted subjective evaluation experiments to determine the suitability of Topic Label and Topical Skewness. I recruited 20 Japanese participants (males and females) because my data sets included Japanese data sets. The ages of the participants ranged from 20 to 30 years old.

I used three data sets to study the correlations between subjective evaluation and automatic evaluation, , i.e., Sankei Sports News, NTCIR3 and NTCIR4 in Table 4.10. I generated three lists of terms. Each of the lists was generated from one of the three data sets as follows. I obtained 100 top-ranked terms by using each of the three methods, i.e., TNG, MI, and RSV. After that, I mixed the three pairs of 100 top-ranked terms into one data set that contained less than 300 terms after eliminating overlaps. Consequently, I obtained three term lists from the three data sets, i.e., Sankei Sports News, NTCIR3 and NTCIR4. I showed the lists of terms sorted in random order to the participants. To each term in the lists, they assigned one of three labels corresponding to the

mixed categories (See Table 4.10). Note that I prepared two additional labels: “uncategorizable” and “unknown term”. The former was for when the participants could not decide on one of the categories of the term because it was ambiguous or corresponded to more than two categories. The latter was for when the participants did not know the meaning of the term. As a result, by checking how many people out of 20 assigned the same labels to a term, I could recognize how much each term leaned towards a topic.

Suitability of Proposed Tools

Table 4.6 lists the correlation coefficients between the number of people who assigned same labels to a term and the value of Topical Skewness, i.e., Equation (4.6), from all data sets. In particular, Figure 4.2 illustrates the results of plotted data from the NTCIR3 data set. The vertical axis represents the number of people who assigned the same labels to a term, and the horizontal axis represents Topical Skewness. Note that I used Pearson’s product-moment correlation coefficient. Table 4.7 shows the number of terms assigned the same label by Equation (4.8) as by subjective evaluation.

In Figure 4.2, each panel represents the result for each label assigned by $TL(t_i)$; i.e., a) is the case when $TL(t_i)$ assigns “憲法，第九条，解釈” (constitution, Article 9, and interpretation), b) is the case when it assigns “京都，寺，神社” (Kyoto, temple, and shrine) and c) is the case when it assigns “著作権，デジタルコンテンツ，ネットワーク” (copyright, digital content, and network). These correspond to the labels of A, B and C in Table 4.10, respectively. Each of the panels has five category signs which represent the label assigned by subjective evaluation, i.e., A, B, C, “uncategorizable” or “unknown term”. The vertical axis represents the number of people who assigned the same label to a term t_i. The horizontal axis is Topical Skewness $TS(t_i)$. Each element represents a term. The rhombus sign represents the case when participants assigned the label of A. The square sign represents the case when participants assigned the label of B. The circle sign represents the case when participants assigned the label of C. In each panel, there is one blacked out category sign. It indicates that the label assigned by subjective evaluation is the same as the label assigned by $TL(t_i)$. In contrast, the outline category signs indicate that these labels assigned by subjective evaluation are different from the label assigned by $TL(t_i)$. Note that if the blacked out category sign shows a larger number of participants than the outline category signs, we can say that $TL(t_i)$ assigned the same labels as assigned by the participants. Additionally, the “×” sign represents the case when

participants assigned the "uncategorizable" label, and the "+" sign represents the case when participants assigned the "unknown term" label. Note that if the blacked out category sign correlates with Topical Skewness, $TS(t_i)$ can be used as a measure of the subjectively assigned topical skewness. In contrast, if the "×" sign inversely correlates with Topical Skewness, we can say that the smaller the value of $TS(t_i)$ is, the higher the ambiguity of t_i becomes.

Accordingly to the above, each panel in Figure 4.2 shows that the blacked out sign presents a larger number of participants than the outline category signs. Most of the outline category signs appear near the zero of the vertical axis. This means that $TL(t_i)$ assigned the same labels as the labels assigned by many participants. Therefore, $TL(t_i)$ can be used instead of subjective labeling. In addition, we can see strong correlations between $TS(t_i)$ and the number of people who assigned the same label of A, B or C. We can also see strong inverse correlations between $TS(t_i)$ and the number of people who assigned the same label of "uncategorizable". Consequently, $TS(t_i)$ can be used as a measure of topical skewness.

Table 4.6 shows the correlation coefficients of subjective evaluation and Topical Skewness. "Number of terms" indicates the number of terms assigned label A, label B, or label C by $TL(t_i)$. For example, the value of 163 in b) NTCIR3 is the number of terms assigned label A by $TL(t_i)$. And the value of 0.65 is the correlation coefficient between $TS(t_i)$ and the number of people who assigned label A to the terms. The boldface lettering in the table indicates that the label assigned by subjective evaluation is the same as the label assigned by $TL(t_i)$. In the same way as described above, if the correlation coefficients are high, $TS(t_i)$ and $TL(t_i)$ can be used as alternatives to subjective evaluation. In addition, if the inverse correlation coefficients are high in the case of "uncategorizable", $TS(t_i)$ can measure the ambiguity of terms. Moreover, statistical significance was assessed with a test of significance of the correlation coefficients. A"*" sign next to the correlation coefficients in Table 4.6 indicates that the correlation test with a significance level of 5% shows a significant positive relationship.

Table 4.6 shows that there are significant positive correlations between the value of Topical Skewness and the number of people who assigned the same label as the label assigned by $TL(t_i)$. In contrast, there are significant negative correlations between Topical Skewness and the number of people who assigned the label of "uncategorizable". Consequently, I can say that $TS(t_i)$ can be used as a measure of the topical skewness that would be assigned by a person.

Table 4.7 shows the number of terms labeled by participants and that by $TL(t_i)$. I tried two ways of counting the terms labeled by the participants: in Case (1), I counted the terms to which more than 10 people assigned the same label, and in Case (2) I counted the terms to which more people assigned the label more often than other labels. For example in case (2), assume that four people assigned A, two people assigned B, one person assigned C, and 13 people assigned "uncategorizable". Because it seems that a term is ambiguous when the majority of people assign "uncategorizable" to it, it is difficult for the labels assigned by $TL(t_i)$ to be the same as the labels assigned by a person. The boldface lettering in the table indicates the number of terms to which $TL(t_i)$ assigns the same labels as those assigned by subjective evaluation. The others are the number of terms to which $TL(t_i)$ assigns different labels from those assigned by subjective evaluation.

Table 4.7 shows that in Case (1), $TL(t_i)$ never assigns labels differently from the labels that more than 10 participants - a clear majority of the 20 participants - assign. That is, when the majority of people associate the same topic with a term t_i, $TL(t_i)$ assigns the proper label to the term. On the other hand, in Case (2), $TL(t_i)$ sometimes assigns labels differently from those of the majority of participants. However, even in this difficult case, $TL(t_i)$ still assigns the same labels as assigned subjectively by some of the participants.

Consequently, we can conclude that $TL(t_i)$ and $TS(t_i)$ can be used as alternatives to human subjective evaluation to assign labels and to measure topical skewness.

4.2.7 Experimental Results and Discussion

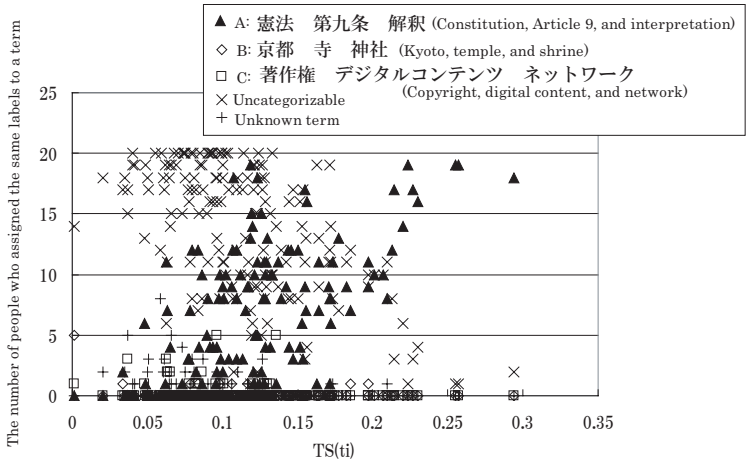
This experiment used $TL(t_i)$ and $TS(t_i)$ described in Section 4.2.5 as alternatives to human subjective evaluation to assign labels and to measure topical skewness.

Extracted Term Evaluation with $TS(t_i)$

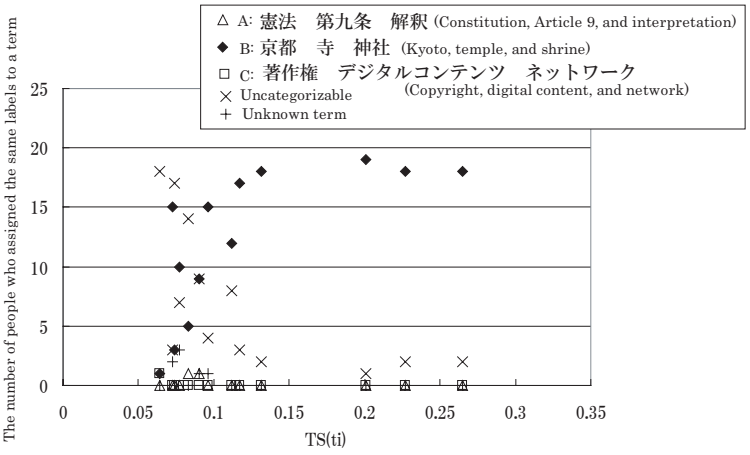
1. On Documents with Three Categories

   I compared TNG with the other term weighting methods, i.e., MI, KLD, χ^2 and RSV. I used the data sets in Table 4.10. Note that I used five different smoothing parameters α (α = 0.05, 0.1, 0.3, 0.5, 1.0) for TNG, MI, KLD and χ^2.

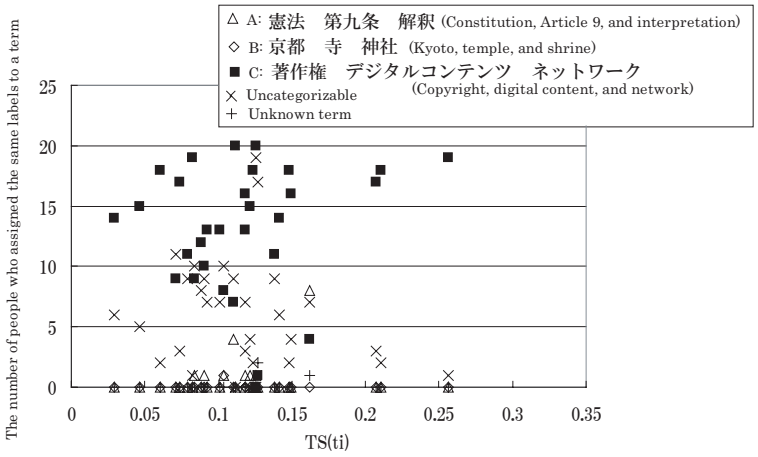
   Figure 4.3 shows the results of the experiment using Reuters. The highest total value of $TS(t_i)$



a)“憲法, 第九条, 解釈” (constitution, Article 9, and interpretation)



b)“京都, 寺, 神社” (Kyoto, temple, and shrine)



c)“著作権, デジタルコンテンツ, ネットワーク” (copyright, digital content, and network)

Figure 4.2: Scattergrams showing correlation between TS evaluation and subjective evaluation of terms obtained from NTCIR3's data and queries.

is achieved when $\alpha = 0.3$. The total value of $TS(t_i)$ is also the highest when $\alpha = 0.3$ in other data sets. Therefore, Figure 4.4 shows the result only for the case of $\alpha = 0.3$. This figure shows the results of the experiment comparing term-weighting methods by using all the data sets. The total value of $TS(t_i)$ of 100 terms top-ranked by each term-weighting method is divided into each category corresponding to the label assigned by $TL(t_i)$ to each term. This experiment did not compare results using different data sets but compared the term-weighting methods on the same data sets.

For all seven data sets, the TNG method gives a higher total value of $TS(t_i)$ in comparison with the other four methods. Moreover, TNG can extract terms corresponding to every topic because the terms it extracts cover all the labels of A, B and C in Figure 4.4. In Table 4.9, each data set has different sizes between document categories. Especially, in the data set of Reuters, the number of documents in category A is nine times that in category C. However, in Figure 4.4, the balance of the total value of $TS(t_i)$ is not influenced by the difference in the number of documents. Therefore, TNG can extract terms regardless of the size difference between categories.

2. On Documents with More Than Three Categories

   This experiment used data sets synthesized by mixing three document categories. I checked whether these data sets were sufficient to examine my method or not. This experiment tried to divide the mixed topics into individual ones. If the synthesized data set was generated by mixing two document categories, e.g., A and B, we cannot distinguish a method to divide them into A and B from a method to divide them into A and not A. Therefore, I mixed three document categories as a minimum setting of the subjective evaluation.

   The retrieved results contained more than three topics despite that the synthesized data sets contained three categories. I also examined the situation with data containing more than three topics by using the NTCIR3 data set. In addition to A, B and C in Table 4.10, I used “ブルーベリー, アントシアニン, 視力” (blueberry, anthocyanin, and vision) as labels of D and “三国志, ゲーム, 題材” (Sanguozhi, game, and subject) as labels of E. The D category contains 188 documents, and the E category contains 167 documents. Figure 4.5 shows the results. In this figure, a) is the case of mixing four document categories from A to D, and b) is the case

Table 4.2: Overall precisions and their improvements compared to the baseline. The baseline overall precision is 0.1606.

method	overall precision	improvement(%)
UnitWeight	0.1765	9.9
TNG1	**0.1847**	15.0
TNG2	**0.1899**	18.2
CF	0.1801	12.1
MI	0.1829	13.9
KLD	0.1733	7.9
χ^2	0.1751	9.0
RSV	**0.1867**	16.3

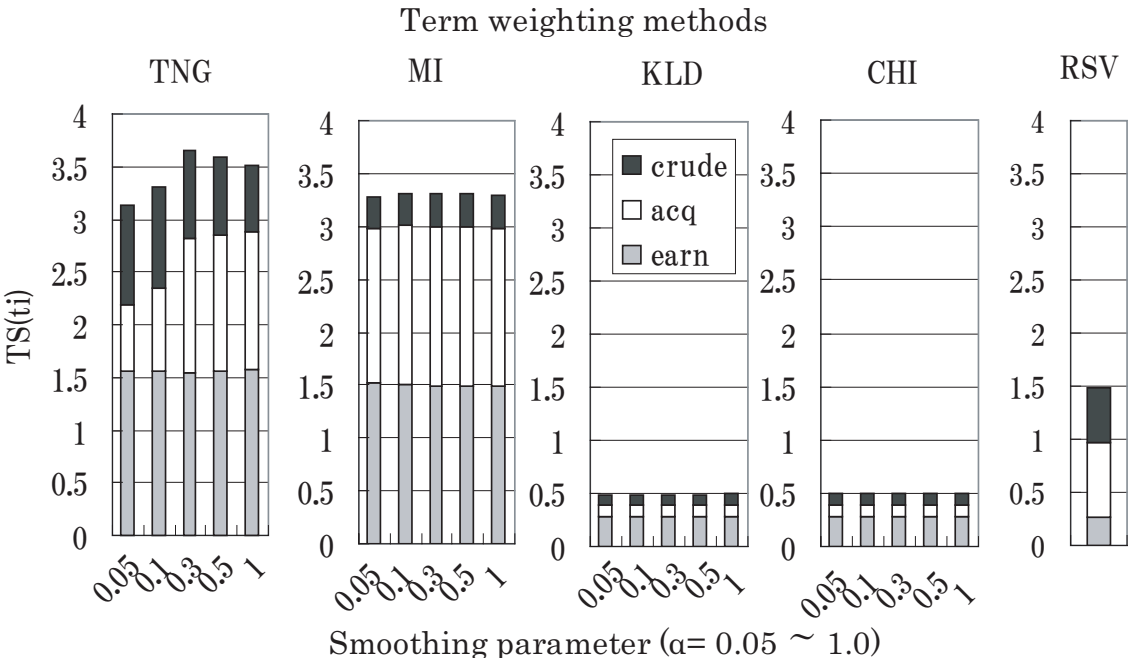


Figure 4.3: Evaluation of term weighting methods by $TS(t_i)$. The terms are obtained from Reuters documents.

Table 4.3:

a) 絶滅＋哺乳類＋危機 (baselin は 0.1291)

手法	上位五語	平均適合率
RSV	種	**0.1212**
	生息	0.1105
	生物	0.0762
	野生	0.0923
	動物	0.0724
TNG1	レッドデータブック	0.0625
	瀬	0.0739
	危惧	0.1680
	危急	0.1027
	シナノミズラモグラ	**0.2361**
TNG2	レッドデータブック	0.0625
	危急	0.1027
	危惧	0.1680
	両生類	0.0747
	ルリカケス	**0.1712**

b) スピーカー+評価+比較 (baselin は 0.0596)

手法	上位五語	平均適合率
RSV	アンプ	0.0067
	可能	0.0279
	結果	0.0341
	システム	0.0246
	音	**0.0593**
TNG1	アンプ	0.0067
	ウーファー	0.0660
	ソフトドームツィーター	0.0154
	スーパーウーファー	0.0177
	バスレフ	**0.0661**
TNG2	アンプ	0.0067
	ウーファー	0.0660
	バスレフ	**0.0661**
	サブウーファー	0.0640
	低音	0.0434

c) 世界樹+北欧神話+名前 (baseline は 0.0675)

手法	上位五語	平均適合率
RSV	神	0.0253
	たち	0.0280
	それ	**0.0377**
	歴史	0.0266
	物語	0.0198
TNG1	Pandaemonium	0.0586
	イグドラシル	**0.3767**
	ソグネフィヨルド	0.0523
	エッダ	0.0525
	シルマリル	0.0533
TNG2	イグドラシル	**0.3767**
	古事記	0.0227
	ギリシア	0.0160
	ノルウェー	0.0203
	フィヨルド	0.0287

Table 4.4: Experimental data and class names or query terms within the data. Terms in Japanese are translated into English.

data set	A	B	C
NTCIR3	憲法 (constitution) 第九条 (Article 9) 解釈 (interpretation)	京都 (Kyoto) 寺 (temple) 神社 (shrine)	著作権 (copyright) デジタルコンテンツ (digital content) ネットワーク (network)
NTCIR4	競馬 (horse racing) 血統 (bloodline)	哲学 (philosophy) 存在論 (ontology)	中国経済 (China economy) 社会主義 (socialism) 市場 (market)
Dmoz	Math	Chemistry	Astronomy
Reuters	earn	acq	crude
Sankei Sports News	Japanese baseball	MLB	soccer
20 Newsgroups	talk.politics.guns	talk.politics.mideast	talk.politics.misc
NTCIR-CLIR	“Give information regarding protests against nuclear power.”	“Articles relating to President Kim Dae-Jung’s policy toward Asia”	“Incidents relating to religious thought about doomsday, or the end of the world.”

Table 4.5: Languages, data types, and numbers of documents used in the experiment. DC: document classification; IR: information retrieval.

data set	language	type	number of documents (A+B+C in Table 4.10)	$\|U\|$ in Equation (4.5)
NTCIR3 web	Japanese	IR	1108(476 + 282 + 350)	10253810
NTCIR4 web	Japanese	IR	2113(643 + 722 + 748)	10253810
Dmoz	English	DC	21089(8935 + 5584 + 6570)	63300
Reuters	English	DC	6615(3845 + 2362 + 408)	9494
Sankei Sports News	Japanese	DC	3519(1233 + 757 + 1529)	10257329
20 Newsgroups	English	DC	3000(1000 + 1000 + 1000)	19955
NTCIR-CLIR	English	IR	209(135 + 50 + 24)	12723

Table 4.6: Correlation coefficient between subjective evaluation and pseudo-evaluation metric $TS(t_i)$.

a) Sankei Sports News

Manually assigned labels	Labels assigned by TL		
	A	B	C
A	**0.85***	0.0039	-0.15
B	-0.034	**0.52***	0.26*
C	0.058	-0.28	**0.76***
Uncategorizable	-0.79*	-0.44*	-0.75*
Unknown term	-0.24	-0.052	-0.019
Number of terms	110	23	92

b) NTCIR3

Manually assigned labels	Labels assigned by TL		
	A	B	C
A	**0.65***	-0.25	0.12
B	-0.16*	**0.69***	-0.051
C	-0.15*	-0.27	**0.14**
Uncategorizable	-0.58*	-0.63*	-0.21
Unknown term	-0.317	-0.40	0.12
Number of terms	163	13	30

c) NTCIR4

Manually assigned labels	Labels assigned by TL		
	A	B	C
A	**0.77***	-0.18	-0.17
B	-0.23	**0.77** *	-0.32
C	-0.16	0.11	**0.84***
Uncategorizable	-0.75*	-0.73*	-0.81*
Unknown term	-0.25	-0.33	-0.34
Number of terms	63	105	37

Table 4.7: Number of terms labeled by users and by TL.

a) Sankei Sports News

Manually assigned labels		Labels assigned by $TL(t_i)$ A	B	C
Case (1): Num. of terms to which more than 10 people assigned the same label	A	**21**	0	0
	B	0	**4**	0
	C	0	0	**25**
	The other case	89	19	67
Case (2): Num. of terms to which more people assigned the label than other labels	A	**56**	9	14
	B	2	**8**	0
	C	6	3	**46**
	The other case	46	3	32
Total number of terms		110	23	92

b) NTCIR3

Manually assigned labels		Labels assigned by $TL(t_i)$ A	B	C
Case (1): Num. of terms to which more than 10 people assigned the same label	A	**35**	0	0
	B	0	**8**	0
	C	0	0	**22**
	The other case	128	5	8
Case (2): Num. of terms to which more people assigned the label than other labels	A	**107**	0	2
	B	2	**12**	0
	C	3	0	**28**
	The other case	51	1	0
Total number of terms		163	13	30

c) NTCIR4

Manually assigned labels		Labels assigned by $TL(t_i)$ A	B	C
Case (1): Num. of terms to which more than 10 people assigned the same label	A	**30**	0	0
	B	0	**18**	0
	C	0	0	**20**
	The other case	33	87	17
Case (2): Num. of terms to which more people assigned the label than other labels	A	**51**	6	1
	B	2	**62**	3
	C	0	13	**28**
	The other case	10	24	5
Total number of terms		63	105	37

of mixing five document categories from A to E.

The results of the experiment using more than three document categories (Figure 4.5) show the same tendency as those using more three document categories. Therefore, it is sufficient to estimate the basic performance by using data sets generated by mixing three document categories. Additionally, the results show that TNG obtains the best performance because the terms extracted by TNG have the highest total value of $TS(t_i)$ and cover all the topics.

Extracted Term Subjective Evaluation

I subjectively compared the performance of TNG with those of MI and RSV. I compared the terms top-ranked by TNG with those of MI and RSV. The smoothing parameter of TNG and MI was set at $\alpha = 0.3$.

Figure 4.6 shows the results. This experiment used three data sets, i.e., Sankei Sports News, NTCIR3 and NTCIR4. I took the ratio of the number of participants assigning the same label to a term to the total number of participants to be the topical skewness of the term. Let $U(L_j, t_i)$ be the number of participants who assign a label L_j to a term t_i. Furthermore, I assumed that label L_j is the label of term t_i when the value of $UT(t_i)$ is the highest of all $L_j (j = 1, 2, 3)$. The topical skewness assigned by subjective evaluation, $UT(t_i)$, is defined as follows:

$$UT(t_i) = \max_j \frac{U(L_j, t_i)}{20} \tag{4.9}$$

Figure 4.6 shows the total value of topical skewness assigned by subjective evaluation.

Figure 4.4 shows that the subjectively evaluated TNG is better than $TS(t_i)$. Hence, the terms extracted by TNG are not only strongly related to topics but also cover all topics exhaustively. Unless the extracted terms cover all topics contained in the document set, some of the topics cannot be shown by the term clusters. Therefore, it is important that the extracted terms cover all topics. Moreover, though it has been the problem that top-ranked retrieved documents are often related to the biggest single topic, I can resolve this problem if the generated clusters cover all topics.

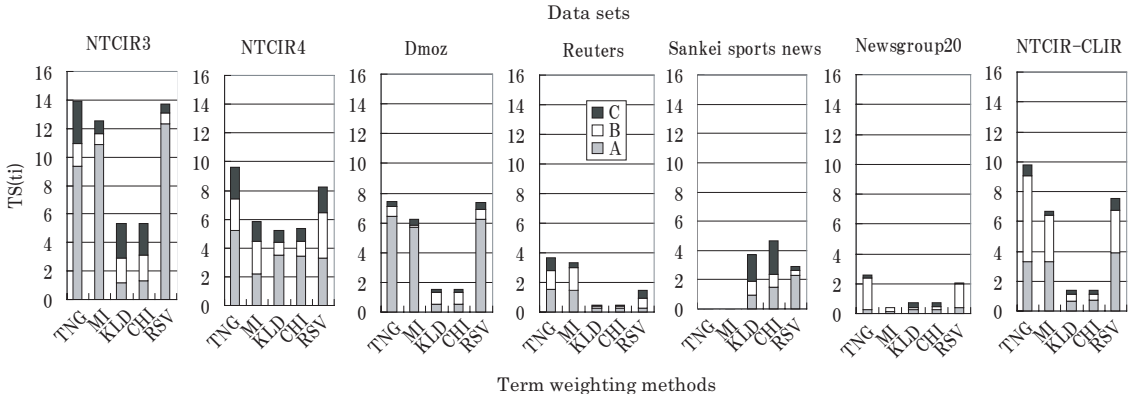


Figure 4.4: Evaluation of term weighting methods by $TS(t_i)$. I used seven different data.

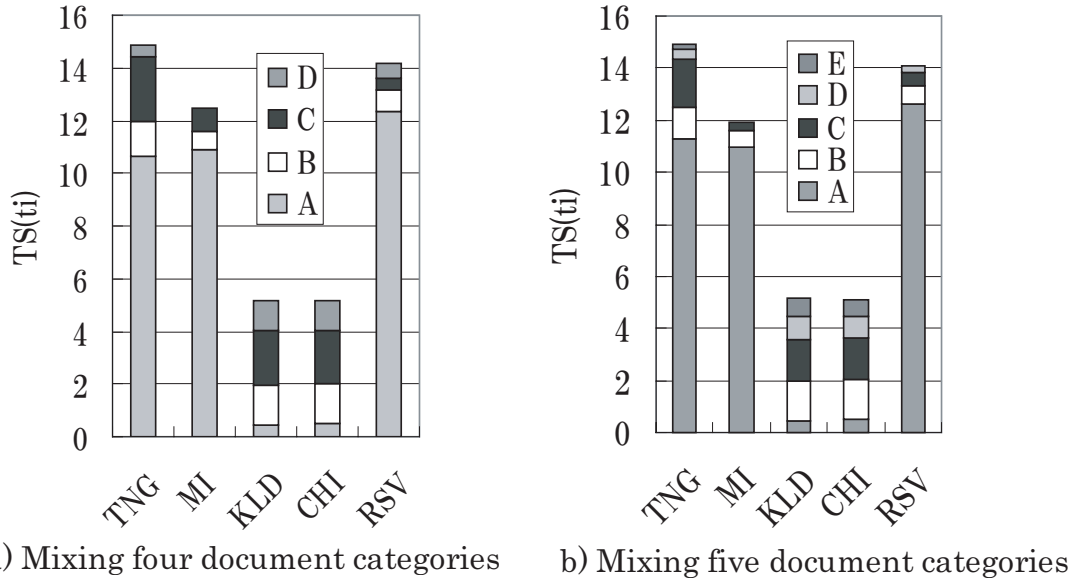


Figure 4.5: Evaluation of term weighting methods by $TS(t_i)$ using pseudo-data containing more than 3 topics of NTCIR3. a)pseudo-data containing four topics and b) pseudo-data containing five topics.

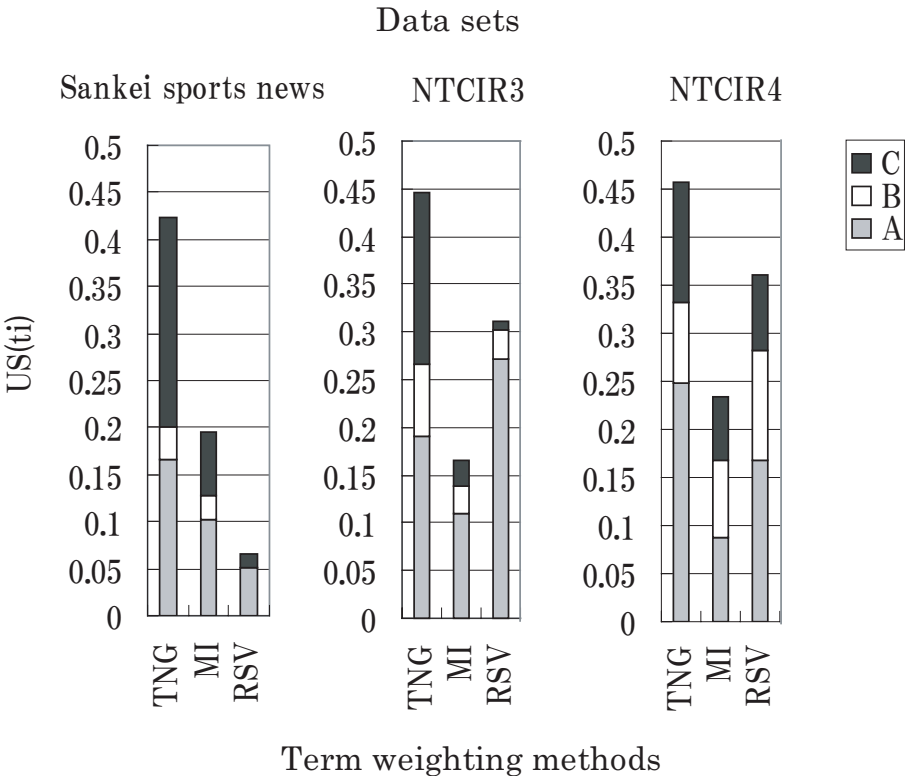


Figure 4.6: Subjective evaluation of terms.

Cluster Evaluation with $TS(t_i)$

I examined the quality of term clusters consisting of terms extracted by each term-weighting method. Let $CTS(C_j)$ denote the topical skewness of a cluster C_j, and let $class_j$ denote the label of the largest number in cluster C_j. $CTS(C_j)$ is defined as follows:

$$CTS(C_j) = \frac{\sum_{t \in class_j} Score(t)}{|C_j|}, \tag{4.10}$$

where

$$class_j = \arg\max_k |\{t | t \in C_j,\ TL(t) = k\}|$$

$$Score(t) = \begin{cases} TS(t) & \text{if } t \in C_j \text{ and } TL(t) = class_j \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

Microaveraged Precision [Cha02] [Seb02] is a measure of the quality of the whole set of clusters. It is easy for this measure to be influenced by the precision of clusters whose sizes are larger than others. Here, based on Microaveraged Precision, I propose a measure defined as $MicroTS$ to be used for the average of $CTS(C_j)$s for all clusters.

$$\begin{aligned} MicroTS &= \frac{\sum_j |C_j| CTS(C_j)}{\sum_j |C_j|} \\ &= \frac{\sum_t Score(t)}{\sum_j |C_j|} \end{aligned} \tag{4.12}$$

The results of my examination of $MicroTS$ are shown in Figure 4.7. All of the smoothing parameters of TNG, MI, KDL and CHI were set at $\alpha = 0.3$. The term clusters generated by using terms extracted by TNG have high $MicroTS$ as does each term before clustering. This means that the term clusters have high precisions and each of them is strongly related to a topic.

Table 4.8 shows examples of actual term clusters made by a) TNG, b) MI and c) RSV. I used the data set generated by mixing three categories “憲法, 第九条, 解釈” (constitution, Article 9, and interpretation), “京都, 寺, 神社” (Kyoto, temple, and shrine) and “著作権, デジタルコンテンツ, ネットワーク” (copyright, digital content, and network) from NTCIR3. I generated term clusters

by using the 100 top-ranked terms gotten by each method. There are 10 term clusters since I set $M = 10$ in Section 3.5.3. We can see that the terms clustered by TNG consist of concrete terms and each of the clusters clearly corresponds to a topic. While RSV obtains high TS in Figure 4.4, the terms it extracted do not comprehensively cover A, B and C. We can see term clusters related to only the topic of "憲法, 第九条, and 解釈" (constitution, Article 9, and interpretation). In contrast, the terms extracted by TNG not only have high TS but also cover all topics comprehensively. As a result, I believe that the term clustered by TNG are better than those clustered by RSV or MI.

I checked which topic each term cluster corresponds to. In the case of a) TNG in Table 4.8, the first cluster seems related to "京都, 寺, 神社" (Kyoto, temple, and shrine), the second and third clusters seem related to "著作権, デジタルコンテンツ, and ネットワーク" (copyright, digital content, and network), and the 4th to 9th clusters seem related to "憲法, 第九条, and 解釈" (constitution, Article 9, and interpretation). In this regard, these term clusters cover all topics comprehensively, although there is a difference between the numbers of clusters related to the same topic. In the future, I believe that the difference between the numbers of clusters related to the same topic can be refined by improving the clustering method. In the case of b) MI, the first cluster seems related to the "京都, 寺, and 神社" (Kyoto, temple, and shrine), clusters from the second to 7th seem related to "憲法, 第九条, and 解釈" (constitution, Article 9, and interpretation), and the 9th cluster seems related to "著作権, デジタルコンテンツ, and ネットワーク" (copyright, digital content, and network). In the case of c) RSV, the clusters seem related to "憲法, 第九条, and 解釈" (constitution, Article 9, and interpretation) but the 8th cluster seems related to "京都, 寺, and 神社" (Kyoto, temple, and shrine).

4.2.8 Summary

This section examined the performance of TNG in comparison with other term weighting methods on multiple data sets. The results showed that TNG can extract terms strongly related to any one of several topics contained in the document set.

Moreover, I proposed a new labeling method that can be used to estimate a topic that each term is related to and the degree of association. The suitability of the labeling was determined through subjective evaluations. Additionally, I evaluated my term weighting method by testing whether the top ranked terms were strongly related to any one of those topics contained in the document set. The

Table 4.8: Examples of term clusters made by a) TNG, b) MI and c) RSV obtained from NTCIR3's data. Each row represents one of the term clusters.

a)TNG

Term Cluster
神社 京都 寺 堂 平安 境内 祭 宮 京 行事
コンテンツ デジタル 著作 音楽 配信 コピー ネットワーク データ 不正 画像 検索 サイト システム
電子 流通 ソフトウェア インターネット 普及 保護 技術 販売 ビジネス アクセス ソフト 町 サービス
武力 報復 テロ 自衛隊 小泉 国連 犠牲 根絶 戦闘
軍事 自衛 軍 憲法 戦争 市 内閣 安保 国民 米 政権
後方 平和 行使 集団 党 首相 攻撃 行動 決議 政党
開発 派遣 選挙 自民党 保障 やる 民主 国会 政治 政府
世論 日本国 議員 九 事態 発言 与党 ブッシュ 反対 戦後 改革 総理 しれる
院 同盟 賛成 湾岸 話 主義 危機 軍隊
輸送 感じる いける

b)MI

Term Cluster
神社 京都 寺 後 市 せる
政府 国民 国会 力 出る 立場 行使 求める 言う 持つ
憲法 戦争 私 平和 状況 協力 軍 ところ とる たち
問題 それ これ 主義 認める 受ける 行為 政策 強い 出す 今回 明らか 参加
国 何 点 くる 以上 いく 人 自由 自衛 判断 主張 安全 数
経済 政治 ない アメリカ 思う 基本 場合 軍事 反対 対応 民主 意味 集団
社会 国際 行動 米 自衛隊 得る 解決 活動 しまう 首相 改正 とき 国連
考える 性 関係 必要 責任 議論 国家 結果 大きい 事態
コンテンツ 著作 デジタル 支援 化 間 可能 委員 具体 いう
本 制度

c)RSV

Term Cluster
憲法 戦争 軍事 平和 テロ 日本 問題 立法 アメリカ 九
権 権利 世界 事態 改憲 主張 時代 保障 守る られる 保護 院 社会事実 的 これ
自衛 国連 武力 行使 小泉 紛争 許す 決議 協力 いう
自衛隊 軍 米 攻撃 集団 国際 軍隊 解決 報復 ため 支援 措置 それ
国会 政府 党 国 安保 反対 政権 内閣 違反 同盟
条 国民 政治 法 外交 放棄 認める 行動 議員 三
首相 行為 自民党 改正 法案 民主 防衛 与党 事件 十 立場
神社 京都 寺
主義 議論 日本国 解釈 条約 国家 侵略 米国 上 著作 重要 自由 政策
手段 法律 政党 諸国

experiment was conducted by using the labeling method for estimating each term's correct topic and by subjective evaluation. The results showed that my method can extract terms strongly related to any one of the topics.

I also evaluated the term clusters obtained by my method. For this, I used an evaluation formula using the labeling method. The results showed that my method can generate term clusters strongly related to each topic. The next section describes experiments on query expansion using such term clusters.

4.3 Experiment for Query Refinement with Topical Term Clusters using TNG

4.3.1 Overview

When we use existing search engines such as Google , Yahoo and MSN , we enter only a few terms to form a query [JSBS98][one]. Then the search engines often return a long list of search results. Even if we use effective query terms, e.g., proper nouns and technical terms, various topics related to the query can be contained in the search results retrieved by such a short query. Therefore, we must select the documents we are interested in from the list by examining the titles and snippets. This is a time-consuming task because the list is unstructured, and it is not easy for web users to understand the multiple topics contained in the search results.

In this work, I propose a method for supporting query refinement by using clusters of topical terms extracted from a retrieved set of documents. I assume that a topic is implied by a specific set of terms that frequently co-occur in the same documents. Therefore, I introduced a new measure of term importance called tangibility (See Section 3.1). A term is said to have tangibility when it frequently co-occurs exclusively with a specific set of terms. Additionally, I proposed two formulae for Tangibility as I mentioned in the Section 3.3.3, that is TNG. Existing methods for term extraction aim to extract terms corresponding to the dominant topic of a given document set, e.g., travel, Europe, and Netherlands. However, because such terms are likely to co-occur with a wide variety of other terms, we cannot distinguish isolated topics. In contrast, my method aims to extract terms exclusively related to one of those topics, e.g., Rembrandt, Schiphol Airport, and canals. Then, I divide the extracted terms into clusters using a distributional clustering algorithm, which leads to agglomerates of terms frequently co-occurring with each other (See Section).

In my experiments, I examined the quality of term clusters by checking effectiveness of the clusters for query refinement. I first extracted topical terms from a synthesized heterogeneous document set, which is a surrogate for a document set retrieved by a real search engine. I then constructed clusters of the extracted terms and used each term cluster as a query to assign a ranking to the documents based on the probabilistic information retrieval model. I used two evaluation criteria: concentration and completeness. First, when the top-ranked documents given by each term cluster

relate to the same topic, we can realize a topic-focusing search by using any one of the constructed clusters. Second, when different term clusters provide different sets of top-ranked documents, we can realize a search covering a wide variety of topics by using all term clusters. I evaluated my term weighting method with these two criteria.

4.3.2 Comparison Methods

In this paper, I compare the proposed method TNG with four other term weighting methods: MI[YP97][YH04], KLD, χ-square [Seb02], and RSV[Rob90]. MI, KLD, and χ-square use term co-occurrence, as does TNG, and these methods can measure how t_j's occurrence probability changes by adding the condition that t_i occurs. Then, the weight of term t_i for these methods is calculated as follows.

$$W(t_i) = \sum_j X(t_j; t_i) \tag{4.13}$$

Here, $X(t_j; t_i)$ is replaced by $MI(t_j; t_i)$, $KLD(t_j; t_i)$, and $\chi^2(t_j; t_i)$ respectively(See Eq. (4.2, (4.3), (4.4)). I also used the same smoothing for the three other methods as for TNG. Moreover, RSV uses the whole document set U, such as the corpus for retrieval, as well as a document subset S, such as search results (See Eq. (4.5)).

4.3.3 Data Set for Experiments

Types of Data My experiments require a document set that includes multiple topics and in which each document has labels indicating a topic. A data set for document classification surely has these labels. Moreover, a test collection for the evaluation of information retrieval is accompanied with relevant document sets for each test query, so we can generate a mixture data set containing multiple topics that are indicated by each relevant document set. Therefore, I used both types of data. Furthermore, I used data sets in Japanese and English to examine the performance regardless of language.

The Data Sets I Used I used seven data sets, described in Table 4.9. I used relevant documents prepared for the Web tasks in NTCIR3 and NTCIR4[EOI+03], documents under the Science

directory in the Dmoz web directory, and Reuters-21578, which was prepared for document classification. I also used back issues of Sankei Sports News on the web. For RSV, I used a corpus of NW100G-01[EOI+03] prepared for NTCIR3 and NTCIR4 as a whole document set U in Eq. (4.5)[††]. I also used Newsgroup20 prepared for document classification in which each category contains about 1000 documents. Finally, I used relevant documents prepared for the tasks of NTCIR-CLIR, which were cross-lingual information retrieval in NTCIR3. These documents were provided by Mainichi Daily News in English.

Each data set has categories, and three of the largest categories are mixed for experiment as pseudo-data including multiple topics. The names of the categories I used are shown in Table 4.10.

The data in Japanese are analyzed by using the MeCab morphological analyzer[MeC] with the ipadic-2.5.1 dictionary[ipa]. The data in English are stemmed by Porter's stemming algorithm, and stop words are then eliminated. I used 1000 of the highest document frequency terms in every data set.

4.3.4 Experimental Procedure

First, I weighted each term using the five methods TNG, MI, KLD, χ-square and RSV for each data set as indicated in Sect. 4.3.3. Next, I made term clusters by using the top 100 terms ranked by each method. Finally, I used every term cluster as a query, and ranked the documents included in the heterogeneous data by using the Okapi probabilistic model[RW99].

Let $Prec(C_i, L_j)$ be the precision when I retrieve documents with a term cluster C_i as a query and use the documents in the category L_j as relevant documents. $Prec(C_i, L_j)$ is defined as the number of retrieved relevant documents in the top x documents divided by x. I define the category $L(C_i)$ corresponding to C_i, as follows.

$$L(C_i) = \arg\max_{L_j} Prec(C_i, L_j) \tag{4.14}$$

[††]In Eq. (4.5), I assume that $|S|$ is 3519, which is the number of all documents included in Sankei Sports News, and $|U|$ is $10,253,810 + 3519$, which is the sum of the number of documents included in NW100G-01 and the number of documents included in Sankei Sports News

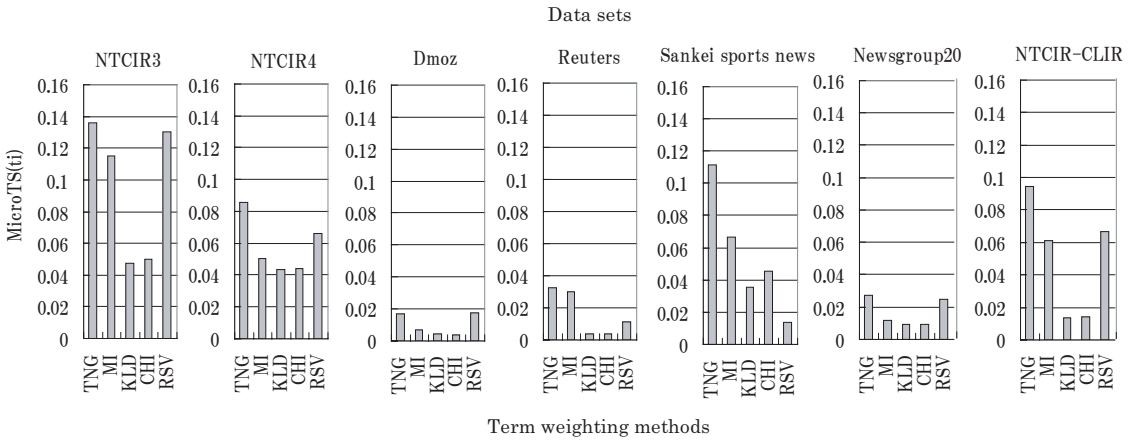


Figure 4.7: Evaluation of term clusters using $MicroTS$. I used seven different data.

Table 4.9: Languages, data types, and numbers of documents I used for the experiment. DC: document classification; IR: information retrieval.

data set	language	type	number of documents (A+B+C in Table 4.10)	$\|U\|$ in Eq. (4.5)
NTCIR3 web	Japanese	IR	1108(476 + 282 + 350)	10253810
NTCIR4 web	Japanese	IR	2113(643 + 722 + 748)	10253810
Dmoz	English	DC	21089(8935 + 5584 + 6570)	63300
Reuters	English	DC	6615(3845 + 2362 + 408)	9494
Sankei Sports News	Japanese	DC	3519(1233 + 757 + 1529)	10257329
Newsgroup20	English	DC	3000(1000 + 1000 + 1000)	19955
NTCIR-CLIR	English	IR	209(135 + 50 + 24)	12723

Table 4.10: Category names or query terms I used. Where the original task is IR, query IDs are shown.

data set	A	B	C
NTCIR3	0032	0013	0028
NTCIR4	0006	0058	0082
Dmoz	Math	Chemistry	Astronomy
Reuters	earn	acq	crude
Sankei Sports News	Japanese baseball	MLB	soccer
Newsgroup20	talk.politics.guns	talk.politics.mideast	talk.politics.misc
NTCIR-CLIR	0036	0023	0018

Further, I define the precision $Prec(C_i)$ of a term cluster C_i, as follows.

$$Prec(C_i) = \max_{L_j} Prec(C_i, L_j) \tag{4.15}$$

Additionally, I define the precision $Prec(L_j)$, which is the maximum precision of $Prec(C_i)$ corresponding to L_j.

$$Prec(L_j) = \max_{C_i \text{ s.t. } L(C_i)=L_j} Prec(C_i) \tag{4.16}$$

First, I examined the relevance of the documents retrieved with the term clusters generated by each method as a query. For this purpose, I compared the average of all $Prec(C_i)$s. Second, I compared the completeness of categories of term clusters by each method by examining the average of $Prec(L_j)$s for all L_j.

4.3.5 Term Cluster Evaluation for Concentrations and Completeness

Comparison of Concentrations I compared the average of $Prec(C_i)$ for all C_i for the top 5, 10 and 100 ranked documents (See Fig. 4.8). TNG had the best average precision for the top 5 ranked documents in all data sets. TNG also had the best average precision for the top 10 ranked documents in NTCIR3, NTCIR4, Dmoz, Sankei Sports News, Newsgroup20, and NTCIR-CLIR. TNG showed stable performance for all data sets and for top 5, 10, and 100 ranked documents. On the other hand, RSV had the second average precision for the top 5 and 10 ranked documents in NTCIR, NTCIR4, Dmoz, NRCIR-CLIR. As a result, TNG outperforms the other comparison methods in the top-ranked documents. Furthermore, TNG outperforms other comparison methods for a wide variety of data sets. That is, TNG extracts terms that are strongly related to one of the three topics.

Comparison of Completeness I compared the completeness of categories for the top 5, 10 and 100 ranked documents (See Fig. 4.9). I evaluated the completeness by the average of $Prec(L_j)$ for all L_j. TNG had the best completeness of categories in NTCIR3, NTCIR4, Dmoz, Sankei Sports News and Reuters. Although χ-square is the best in Newsgroup20 and RSV is the best in NTCIR-CLIR, these methods did not outperform TNG in the other data sets. Therefore, TNG outperformed

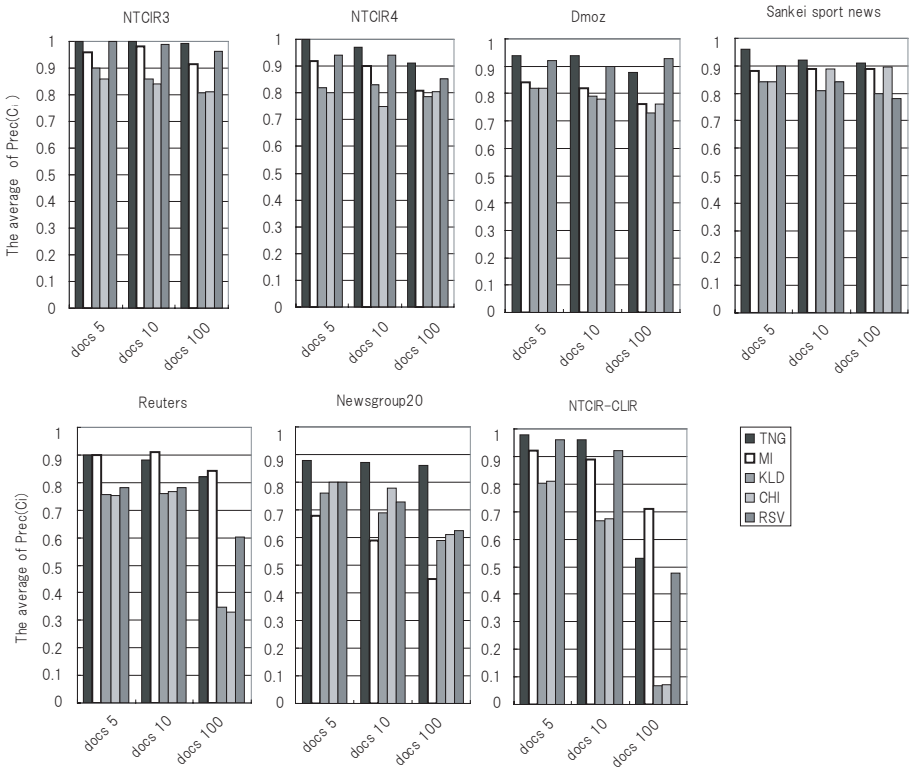


Figure 4.8: The average of $Prec(C_i)$ for all C_i for the top 5, 10 and 100 ranked documents of NTCIR3, NTCIR4, Dmoz, Sankei Sports News, Newsgroup20, and NTCIR-CLIR.

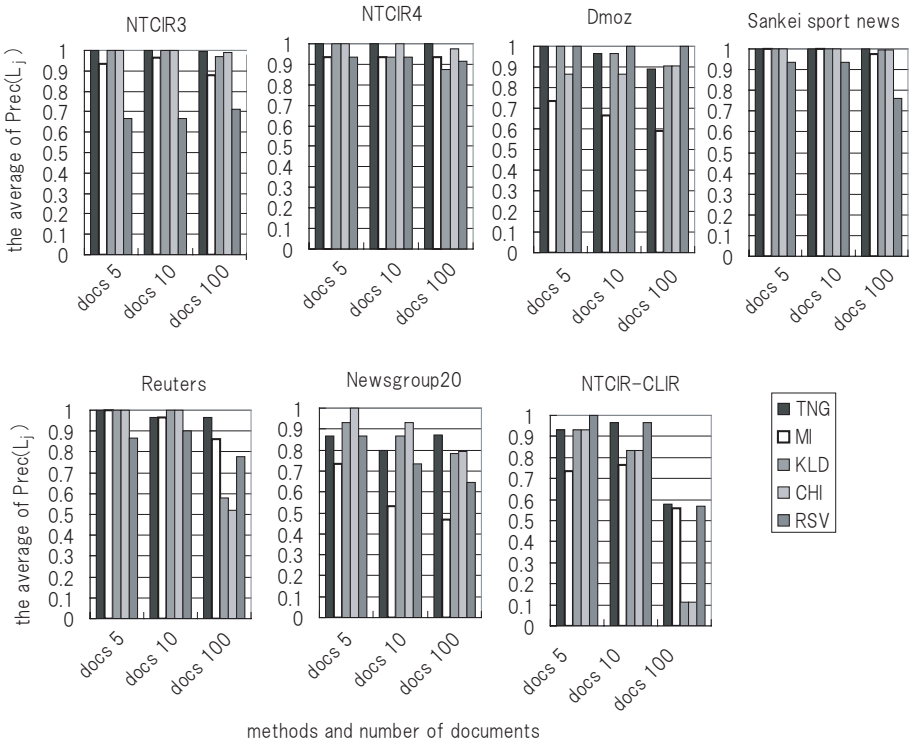


Figure 4.9: The average of $Prec(L_j)$ for all L_j for the top 5, 10 and 100 ranked documents of NTCIR3, NTCIR4, Dmoz, Sankei Sports News, Newsgroup20, and NTCIR-CLIR.

other methods with respect to overall performance.

4.3.6 Summary

This section examined the performance of TNG in comparison with other term weighting methods on multiple data sets. First, I extracted terms using each method and generated term clusters by using these terms. Next, I retrieved documents with the term clusters as a query from a heterogeneous set of documents. With respect to the average precision of documents retrieved by the clusters, TNG outperformed other methods. Furthermore, TNG had a good completeness of categories of documents retrieved by the term clusters. We can conclude that TNG is an efficient term weighting method for detection of topics included in a heterogeneous set of documents. I think that TNG can be used for query expansion that considers that various topics are contained in the first-retrieved documents and that users can select one of those topics efficiently with my method.

Chapter 5

Representation System of Multiple Topics with Search Results: Application of Tangiblity

5.1 Overview

When we use search engines such as Google, Yahoo and MSN, we enter only a few terms to form a query [JSBS98]. The search engines often return a long list of search results. Even if we use effective query terms, e.g., proper nouns and technical terms, a short query may cause the search engine to0 retrieve search results on various topics. In such a situation, the user must select the documents he or she is interested in from the retrieval list by examining the titles and snippets. This is a time-consuming task because the list is unstructured, and it is not likely easy for web users to understand all the topics contained in one set of search results.

A number of search engines that organize retrieved results have been developed recently. For instance, Clusty* is a document clustering search engine. Clusty also shows the labels of document clusters to searchers. Another example is "Google suggest," which uses user logs to show queries to refine the original query. The terms presented by these systems are likely to be general terms because they regard terms that many searchers used or terms that retrieve a large amount of documents as

*http://clusty.com/

being useful. Moreover, these terms are not organized based on clear concepts.

I proposed a method for supporting query refinement by using clusters of topical terms extracted from a retrieved set of documents. I assume that a topic is implied by a specific set of terms that frequently co-occur in the same documents. Therefore, I introduced a new measure of term importance called *tangibility* and a method for calculating it called TNG (See Section 3.3.3). A term is said to have tangibility when it frequently co-occurs exclusively with a specific set of terms (See Section 3.1). I divide the extracted terms into clusters by using improved distributional clustering algorithm, which leads to agglomerates of terms frequently co-occurring with each other (See Section 4.3.1).

In this chapter, I develop a system using proposed TNG and generating term clusters to show topical term clusters that can be used for query refinement. I also subjectively evaluated the performance of TNG. The results show that the term clusters generated by my system using TNG can distinguish the topics of the search. Moreover, the term clusters help us to find multiple meanings of the query and to discover unexpected topics related to it.

5.2 System Architecture

I developed a system using TNG described in Section 3.3.3 to show topical term clusters that can be used for query refinement. My system is implemented in the Ruby programming language, CGI in Perl, and wget. I used MeCab[†] as the part-of-speech (POS) and morphological analyzer for Japanese. The dictionary of MeCab was ipadic-2.5.1[‡].

First, this system gathers the top 500 URLs as ranked by Google for a query given by the user. Then it removes the dead links and the URLs referring to non-HTML documents. By removing HTML tags, I get the raw text of each document. As the downloaded documents show a wide divergence in their lengths, I extract snippets (i.e., short summaries of documents) of constant length from each document. My method for snippet extraction is described in Section 5.2.1. The system calculates TNG scores by regarding each snippet as a single document and generates term clusters with my improved distributional clustering algorithm (See Section 4.3.1). I provide the details of this clustering phase in Section 5.2.2. The user can use each term cluster as additional query terms to expand the original query. Figure 5.1 depicts the system architecture.

[†]http://mecab.sourceforge.jp/
[‡]http://chasen.naist.jp/stable/ipadic/

5.2.1 Extraction of Snippets

Retrieved documents show a wide divergence in their lengths and also have a wide variety of terms. To use only the area related to the query, I extracted snippets from each document as follows[§].

1. A sliding window 30 terms long was superimposed over one text stream and shifted.
2. I counted the frequency of the query terms (FQ), the number of unique queried terms (NQ), and the number of nouns (NN) inside every window.
3. The window was shifted until one reached the end of the text stream, after which all of the windows were sorted according to the valance of FQ, NQ, and NN.
4. The two highest scored windows were selected as the snippet of the current text.

Note that I used the text inside the first 1 Mbyte until less than 10 query terms appeared. I also removed stop words, e.g., well-known stop words in English, common terms in Japanese, and common terms in HTML documents either in English or Japanese.

5.2.2 Applying the Proposed Method

Using snippets in Section 5.2.1, I calculated the TNG scores proposed in Section 3.3.3. I used the top 500 terms sorted by document frequency. Next, I used the top 200 terms sorted by TNG for term clustering (See Section 4.3.1). Note that I changed step 4 (i) of the clustering algorithm so the number of clusters was automatically fixed. Instead of the step 4 (i), I merged all pairs of clusters whose similarity was more than a similarity threshold set at 0.01. Note that I say two terms co-occur when they both appear in the same snippet. I ignored only one co-occurrance as noise. I set the number of clusters M to 20. Though the number of clusters was automatically fixed by the algorithm, I used the top 10 term clusters sorted by TNG score when I assumed that each score was the highest among the terms contained in each cluster. The terms contained in the term clusters were sorted by the document frequency, and I used the top 5 subjectively evaluated terms (See Figure 5.2).

[§]I conducted two pre-experiments. One of them was an experiment using snippets extracted by Google as documents. The other was an experiment using whole texts as documents. The former was better than the latter. This is because a whole text usually contains many topics. And retrieved documents show a wide divergence in their lengths and also have a wide variety of terms. Therefore, I extracted snippets of constant length.

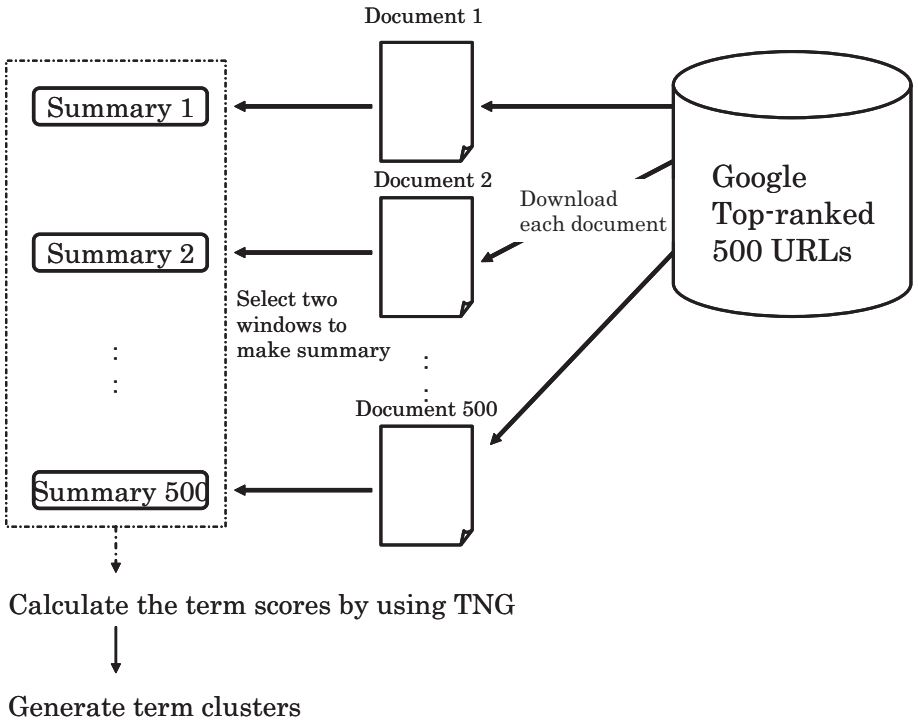


Figure 5.1: Whole process of system using TNG.

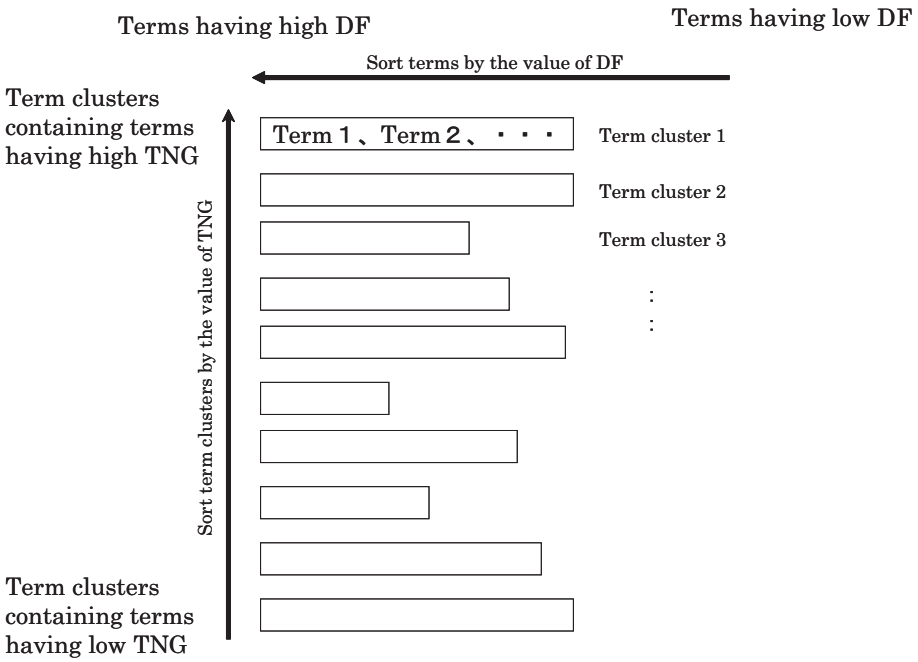


Figure 5.2: Sorting method for term clusters and terms contained in them to show the participants

5.2.3 Examples of System Output

Table 5.1 and Table 5.2 show the top 10 term clusters output by my system. Table 5.1 shows the term clusters for the query “ジャガー”, and Table 5.2 shows the term clusters for the query “ニンテンドー DS”. In Table 5.1, for example, Cluster 1 indicates the cars of Jaguar Corporation because there are brands of cars. Cluster 2 indicates ジャガー横田 (Jaguar Yokota[¶]) because there are the names of her and her husband and some topical terms strongly related to her. In the same way, Cluster 3 indicates Jaguar xkr models. Cluster 4 indicates one of the ‘big cats’, i.e., the jaguar. Cluster 5 indicates a manufacturer of sewing machines. Cluster 6 indicates used cars of the Jaguar Corporation. Cluster 8 indicates jaeger lecoultre, a brand of wrist watch. Cluster 10 indicates the latest models of Jaguar cars.

5.3 Performance of My System

5.3.1 Compared System

I compared my method with Clusty[‖]. Clusty generates document clusters to organize the search results for a query given by the user. Clusty also shows the labels of the document clusters. I used these labels as additional terms to expand the original query. Clusty may show a set of two or more terms as a label of a document cluster. In this case, I used all of these terms to expand the original query. The labels given by Clusty were compared with the term clusters generated with TNG. Since Clusty shows 10 labels in the default configuration, I used all 10 labels generated by Clusty and the top 10 term clusters by TNG.

5.3.2 Participants

I recruited 8 male participants. The participants worked in economics and computer science fields of information retrieval, communications networks, and computer graphics. All of the participants have used search engines for 6 to 10 years. The ages of the participants ranged from 23 to 30 years. As their main search engines, eight participants used Google, two used Yahoo!JAPAN, two used MSN, and one used goo when multiple answers were allowed. Six participants usually used

[¶]“ジャガー横田”is a famous female Japanese pro-wrestler.
[‖]http://clusty.jp/

Table 5.1: Term clusters generated by our system for the query “ジャガー’ (jaguar). Terms translated into English are shown in parentheses.

ID	Term cluster
1	フォード (Ford) トヨタ (Toyota) BMW ポルシェ(Porsche) アウディ(Audi)
2	横田 (Yokota) 出産 (birth) 女子 (girl) 木下 (Kinoshita) プロレスラー (pro wrestler)
3	new 注文 (order) 搭載 (equipped) 開始 (beginning) XKR
4	動物 (animal) ネコ (cat) アメリカ (United States) 妊娠 (pregnancy) 食肉 (predatory)
5	ミシン (sewing machine) 説明 (explanation) net ロック (overlock) 電子 (electronic)
6	中古 (used) 輸入 (import) ディーラー (dealer) 正規 (regular) 認定 (recognition)
7	買取 (purchase) 記入 (fill-in) 高額 (more money) 完了 (completion) 必須 (indispensability)
8	ルクルト (lecoultre) 腕時計 (wristwatch) lecoultre jaeger
9	内容 (content) 著者 (author) 投票 (vote) 書名 (book title)
10	車種 (model of car) 最新 (latest) 使用 (use)

Table 5.2: Term clusters generated by our system for the query “ニンテンドー DS” (Nintendo DS). Terms translated into English are shown in parentheses.

ID	Term cluster
1	発表 (announcement) ドラゴンクエスト (Dragon Quest) ドラクエ (abbr. of Dragon Quest) ドラゴン (Dragon) 新作 (new work)
2	アクション (action) シミュレーション (simulation) ロールプレイング (role playing) アドベンチャー (adventure) パズル (puzzle)
3	プレイステーション (PlayStation) xbox(Xbox) キューブ (Cube) ゲームボーイアドバンス (Game Boy Advance) ボーイ (boy)
4	常識 (Common Sense) 監修 (under the supervision) モンスター (monster) 検定 (test) ダイヤモンド (diamond)
5	任天堂 (Nintendo) nintendo 関連 (relevant) amazon クリスタルホワイト (crystal white)
6	無線 (wireless) lan usb コネクション (connection) 接続 (connection)
7	開発 (development) opera 共同 (joint) software
8	拡張 (extended) カートリッジ (cartridge) メモリー (memory)
9	ブラック (black) ホワイト (white) クリスタル (crystal) ジェット (jet)
10	トピックス (TOPIX) 経済 (economy) ウェブ (web) 地域 (area) floor

three or fewer query terms, one used five or fewer, and one used ten or fewer. In addition, five participants said they usually checked the top thirty pages or less, and the other three participants said they usually checked one hundred, two hundred, and three hundred pages or less, respectively.

5.3.3 Queries

I used twenty queries in Table 5.3. Table 5.3 shows not only the query terms I used but also their sources. Eight of them come from search term rankings by Yahoo!Japan [**][††] The list contains such queries as "mixi" in the first place of the all-around ranking, "KAT-TUN" in the first place of the popularity ranking for males, "あいのり" in the first place of the popularity ranking for TV programs, "DEATH NOTE" in the first place of the popularity ranking for comics and cartoons, and "ニンテンドー DS" (Nintendo DS) in the first place of the popularity ranking for commercial products. Additionally, the list contains such queries as "ジャガー" (jaguar) and "アップル" (apple) that are often used as polysemic words. Moreover, it contains query terms used in the third NTCIR web retrieval task [EOI+03] and some general terms.

5.3.4 Pre-Experiment Questionnaire

Before the start of the experiment, I gave a questionnaire about the twenty queries to the participants. Table 5.4 lists questionnaire items q1 to q4. Note that the participants answered the questionnaire either with Y (yes) or N (no) for q1, q3, and q4, and they answered Y (yes), N (no), or M (medium) for q2. The number in Table 5.4 would be 8 (participants) ×20 (queries) = 160 if all participants answered Y. Table 5.4 shows that most participants had heard of the queries before, and they thought they knew what most of the queries meant. Therefore, we can say that all participants had background knowledge on all the queries to some extent. In addition, they recognized 6.5 words as polysemic words and 15 words as words related to multiple topics within the 20 queries.

5.3.5 Experimental Procedure

1. Retrieval Experiment using only Term Clusters

   The participants checked whether the term clusters retrieved documents related to the original

[**] http://picks.dir.yahoo.co.jp/new/review2006/general.html
[††] Counted from Jan. 1st to Nov. 5th in 2006.

Table 5.3: Queries I used and their sources. (“(*)” means that the query is from Yahoo!Japan.)

Label	Query	Source
A	飛行機 (airplane)	General term
B	北朝鮮 (North Korea)	News
C	アムステルダム (Amsterdam)	Name of a famous place
D	無免許 (unlicensed)	News
E	クラスタリング (clustering)	Technical term
F	野村 (Nomura)	General name in Japanese
G	情報 (information)	Having multiple meanings
H	アップル (apple)	Polysemic word
I	ジャガー (jaguar)	Polysemic word
J	KAT-TUN (KAT-TUN)	The first place in popularity ranking for males (*)
K	ニンテンドー DS (nintendo DS)	The first place in popularity ranking for commercial products (*)
L	mixi (mixi)	The first place in all-around ranking (*)
M	あいのり (Ainori)	The first place in popularity ranking for TV programs (*)
N	涼宮ハルヒの憂鬱 (The Melancholy of Haruhi Suzumiya)	Popular search trend (*)
O	ワンセグ (1seg)	Popular search trend (*)
P	成分分析 (componential analysis)	Popular search trend (*)
Q	DEATH NOTE (DEATH NOTE)	The first place in popularity ranking for comics and cartoons (*)
R	哺乳類 絶滅 (mammal, extinction)	NTCIR3 Web retrieval task
S	北欧神話 世界樹 (Norse mythology, World Tree)	NTCIR3 Web retrieval task
T	レオナルド・ダ・ヴィンチ (Leonardo da Vinci)	Name of a famous person

query shown in Table 5.3 without using the query. If the query terms were expanded with the term cluster, it is not surprising that the documents related to the query were retrieved. In contrast, if the term cluster retrieved documents related to the original query without using the query, it must help to focus on the query. This is why I tried to determine whether term clusters without the original query can retrieve the documents related to the original query.

The experimental details are as follows. I placed the document cluster labels of Clusty on the left and term clusters generated by my system using TNG on the right of Figure 5.3. Note that the participants did not know what system generated each of the labels and the term clusters. The participants used each of 10 labels given by Clusty to expand the original query. By browsing the Google search results obtained by the expanded query, they checked for the search results related to the original query. When the search results gotten by using the label as an additional query were not related to the original query, they removed it from the space. Regarding TNG, the participants used each of 10 term clusters generated by my system using TNG to expand the original query and browsed the search results. When the search results gotten by using the term cluster as an additional query were not related to the original query, they removed it.

Next, they used each of the remaining document labels given by Clusty for the search without the original query terms. If one of the document labels retrieved documents related to the original query, they marked the document label as having focusing ability on a topic. If one of the document labels did not retrieve the documents related to the original query, they did not mark the document label. Note that I did not eliminate the query terms from the term cluster if it contained the original query terms when Clusty output it as being useful. For my system using TNG, they used each of the remaining term clusters for the search without the original query terms. If one of the term clusters retrieved the documents related to the original query, they marked the term cluster as having focusing ability on a topic. If one of the term clusters did not retrieve the documents related to the original query, they did not mark the term clusters.

The focusing ability of the document labels denoted by $FA_{Clusty}(q)$ is defined as follows.

$$FA_{Clusty}(q) = N_{retrieve}/N_{remaining} \quad (5.1)$$

where q denotes the original query, $N_{remaining}$ denotes the number of remaining document labels in the space, and $N_{retrieve}$ denotes the number of document labels marked by the participants. For example, assume that there are eight out of ten remaining document labels when the query term is q_1. Further, assume that there are three out of ten document labels that are able to retrieve the documents related to the original query without the original query. In this case, I can calculate $FA_{Clusty}(q_1)$ as three eighths.

The same as the document labels, I can also calculate the focusing ability of the term clusters. The focusing ability of the term clusters $FA_{TNG}(q)$ is calculated as

$$FA_{TNG}(q) = N_{retrieve}/N_{remaining} \quad (5.2)$$

where q denotes the original query, $N_{retrieve}$ denotes the number of term clusters marked by the participants, and $N_{remaining}$ denotes the number of remaining term clusters in the space.

2. Experiment on Usability of Term Clusters for Query Expansion

   The participants compared my system using TNG with Clusty by using the procedure described below. The participants used each of 10 labels given by Clusty to expand the original query. By browsing the Google search results obtained by the expanded query, they could check the effectiveness of the document cluster labels of Clusty. As for TNG, the participant used each of 10 term clusters generated with TNG to expand the original query and browsed the search results. Note that I placed the document cluster labels of Clusty on the left and term clusters generated by my system using TNG on the right of Figure 5.3. Note that the participants did not know what system generated each of the labels and the term clusters.

   This procedure was repeated for all 20 queries in Table 5.3. With respect to each query, the participants were required to answer three yes-or-no questions.

   - Q1: Have you found multiple meanings related to the query?

- Q2: Have you found unexpected topics related to the query?
- Q3: Did these term clusters enhance your knowledge about the query?

The participants answered these questions for both the set of 10 labels given by Clusty and the set of 10 term clusters given by TNG. Therefore, the number for Clusty would be 8 (participants) $\times$20 (queries) = 160 if all participants answered Y.

They answered Q4 and Q5:

- Q4: Which term cluster set do you think is more convenient to search with?
- Q5: Which term cluster set do you think is more convenient to find interesting topics with?

Through 20 queries, they answered the above questions by either on the left terms or the right terms of the space. Therefore, the number would be 8 (participants) if all participants answered left.

5.3.6 Results

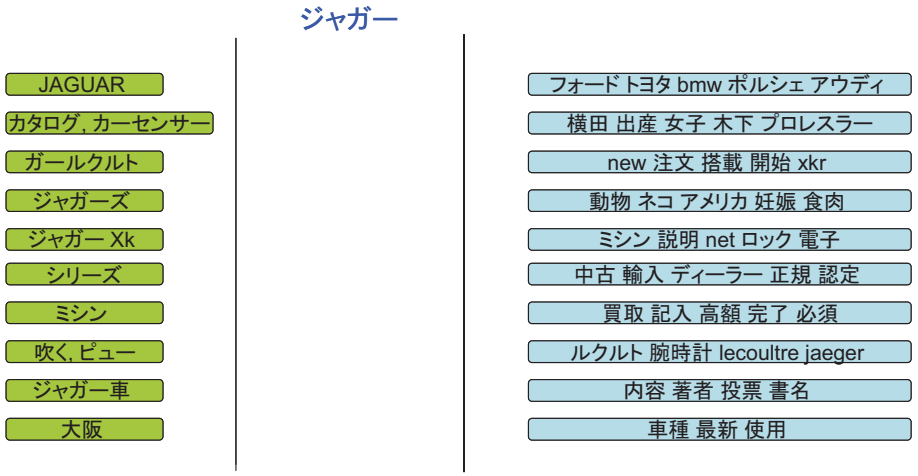
1. Results of Retrieval Experiment using only Term Clusters

   First, I computed the focusing ability of the document labels generated by Clusty by Equation (5.1) for each of the eight participants. I also computed the focusing ability of the term clusters generated by my system using TNG by Equation (5.2) for each of the eight participants. Then, I computed the average of the eight focusing ability values for both Clusty and TNG. I repeatd this computation for each of the 20 queries. With respect to the query q, we denote the average of the focusing ability values for Clusty by $\mu_{Clusty}(q)$, and denote the average of the focusing ability values for TNG by $\mu_{TNG}(q)$.

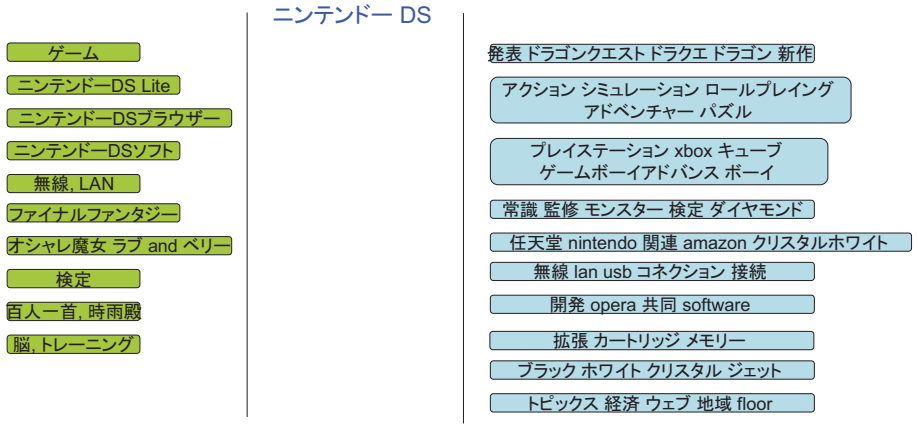
   Figure 5.4 shows the results of the balance between the focusing ability values of TNG and Clusty. I plotted the points $(x, y) = (\mu_{Clusty}(q), \mu_{TNG}(q))$ for each query q. The horizontal axis represents the value of $\mu_{Clusty}(q)$. The vertical axis represents the value of $\mu_{TNG}(q)$. The statistical significances of the average values in the figure are shown in Table 5.7. Moreover, the statistical significance was assessed with a test of significance by t-test. Table 5.7 lists t-value, p-value, and degree of freedom. Seventeen queries out of 20 show averages with

Table 5.4: Results for q1 to q4. (The average number of queries per participant is shown in ().)

Question	Y	M	N
q1: Have you heard of the query before?	156 (19.5)	-	4 (0.5)
q2: Do you think you know about the query?	36 (4.5)	96 (12.0)	28 (3.5)
q3: Do you think the query has multiple meanings?	52 (6.5)	-	108 (13.5)
q4: Do you think the query is related to multiple topics?	119 (14.9)	-	41 (5.1)



a)Result for the query “ジャガー” (jaguar).



b) Result for the query “ニンテンドー DS” (Nintendo DS).

Figure 5.3: Term clusters presented to the participants. The term clusters on the left were generated by Clusty, and those on the rightwere generated by my system. The term clusters generated by Clusty are translated into English on Table 5.5 and 5.6. The term clusters generated by my system are translated into English on Table 5.1 and 5.2.

Table 5.5: Document labels generated by Clusty for the query “ジャガー” (jaguar).

ID	Term cluster
1	JAGUAR
2	カタログ カーセンター (catalog car center)
3	ガールクルト
4	ジャガーズ (jaguars)
5	ジャガー Xk(jaguarXk)
6	シリーズ (series)
7	ミシン (sewing machine)
8	吹く (blow) ピュー (whiz)
9	ジャガー車 (jaguar car)
10	大阪 (Osaka)

Table 5.6: Document labels generated by Clusty for the query “ニンテンドー DS” (Nintendo DS).

ID	Term cluster
1	ゲーム (game)
2	ニンテンドー DS Lite(Nintendo DS Lite)
3	ニンテンドー DS ブラウザ (Nintendo DS browser)
4	ニンテンドー DS ソフト (Nintendo DS soft)
5	無線 LAN (wireless LAN)
6	ファイナルファンタジー (Final Fantasy)
7	オシャレ魔女 ラブ and ベリー (Oshare Majo: Love and Berry)
8	検定 (test)
9	百人一首 (Hyakunin isshu) 時雨殿 (Shigureden)
10	脳 トレーニング (Brain Training)

significance levels of 5%, 1%, or 0.1%. The diagonal line connecting the point $(0,0)$ to $(1,1)$ represents the case when the document labels generated by Clusty and the term clusters generated by my system showed the same efficiency in retrieving documents related to the query as without using the query. If a query is on the left above the line, the term clusters generated by my system are more efficient for retrieving documents related to the query than the document labels generated by Clusty. In contrast, if a query is on the right under the line, the document labels generated by Clusty are more efficient for retrieving documents related to the query than the term clusters generated by my system. All the queries are on the upper left above the line except for "アムステルダム"(Amsterdam), "野村"(Nomura), "アップル"(apple), and "ジャガー"(jaguar). In these cases, we can say that my system using TNG gave term clusters a stronger ability to focus on the topics related to the query than the document labels generated by Clusty. We can recognize that the four queries on the lower right have multiple meanings. We discuss this point from the statistical viewpoint in the next paragraph.

On the other hand, we know how many people answered yes to q3 in the pre-experimental questionnaire for each of the 20 queries. With respect to the query q, I denote the number of people who answered yes to q3 by $\nu(q)$. Then, I computed the correlation coefficient between the 20 values $\mu_{Clusty}(q_A), \ldots, \mu_{Clusty}(q_T)$ and the 20 values $\nu(q_A), \ldots, \nu(q_T)$ for Clusty. I also computed the correlation coefficient between the 20 values $\mu_{TNG}(q_A), \ldots, \mu_{TNG}(q_T)$ and the 20 values $\nu(q_A), \ldots, \nu(q_T)$ for TNG. Note that I used Pearson's product-moment correlation coefficient. Moreover, I assessed the statistical significance. The results are shown in Table 5.8. I got $r = .48, T(18) = 2.3, p < .05$ for Clusty, and $r = -.46, T(18) = 2.2, p < .05$ for TNG . There are strong correlations between the average $FA_{Clusty}(p)$ and the number of people who recognized the query as having multiple meanings. In addition, there are strong inverse correlations between the average $FA_{TNG}(p)$ and the number of people who recognized the query as having multiple meanings. Consequently, we can conclude as follows for the search without using the original query. When the query has multiple meanings, Clusty's document labels retrieve the documents related to the query more efficiently than TNG's term clusters. On the other hand, when the query has a unique meaning, TNG's

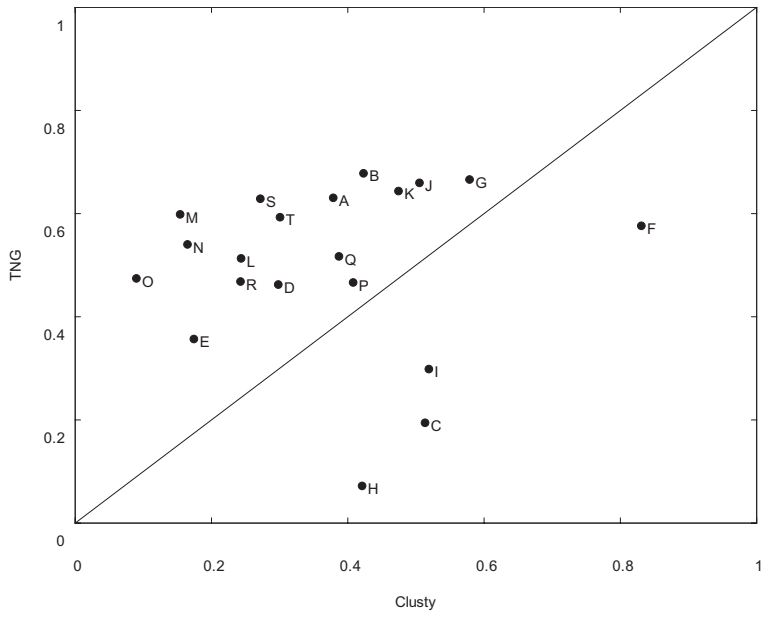


Figure 5.4: Comparison between the focusing ability values of TNG and that of Clusty. The plotted points are $(x, y) = (\mu_{Clusty}(q), \mu_{TNG}(q))$ for each query q. See Table 5.3 for the labels assigned to the points correspoinding to the queries.

term clusters retrieve documents related to the query more efficiently than Clusty's document labels. Therefore, we can say that the four queries on the right and under the line in Figure 5.4 must have multiple meanings.

2. Results of Experiments on Usability of Term Clusters for Query Expansion

   Table 5.9 and Table 5.10 show the results of Q1 to Q5 mentioned in Section 5.3.5. Table 5.9 shows the number of participants who answered Q1 to Q3 by Y for the document labels generated by Clusty and the term clusters generated by TNG. It also shows the averages for all eight participants. Moreover, statistical significance was assessed with a test of significance by t-test for the Q1 to Q3. The t-value, p-value, and degree of freedom are also shown in the table. The answers to Q1 show that the participants found multiple meanings by using term clusters of both TNG and Clusty because there is no statistical significance in the difference between the averages. Although the result of the pre-experiment questionnaire indicated that the participants recognized an average of 6.5 words as polysemic words out of 20 queries, they recognized an average of 10.5 words as polysemic words after the experiment. Therefore, we can say that the participants could easily recognize multiple meanings of the query after

Table 5.7: The statistical significances of the average values of $\mu_{Clusty}(q)$ and $\mu_{TNG}(q)$. The labels corresponding to queries are shown in Table 5.3.

Label	t-value and p-value
A	$T(7) = 3.5, p < .01$
B	$T(7) = 4.2, p < .01$
C	$T(7) = 3.0, p < .01$
D	$T(7) = 1.9, p < .05$
E	$T(7) = 2.3, p < .05$
F	$T(7) = 2.5, p < .05.$
G	$T(7) = 1.5, p = n.s.$
H	$T(7) = 5.6, p < .001$
I	$T(7) = 6.4, p < .001$
J	$T(7) = 3.5, p < .01$
K	$T(7) = 2.4, p < .05$
L	$T(7) = 5.0, p < .001$
M	$T(7) = 4.2, p < .01$
N	$T(7) = 7.1, p < .001$
O	$T(7) = 3.9, p < .01$
P	$T(7) = 1.3, p = n.s.$
Q	$T(7) = 1.8, p = n.s.$
R	$T(7) = 5.1, p < .001$
S	$T(7) = 5.6, p < .001$
T	$T(7) = 6.6, p < .001$
Average of 20 queries	$T(7) = 6.7, p < .001$

Table 5.8: Correlation coefficient between average $P_{method}(q)$ of eight participants and the number of people who answered yes to question q3 in the pre-experimental questionnaire. (*method* denotes Clusty and TNG.)

Statistical test	Clusty	TNG
correlation coefficient	0.48	-0.46
t-value	2.3	2.2
p-value	0.016	0.021

being shown multiple topics related to the query because the number of words recognized as polysemic words increased.

The number of queries to which the participants found unexpected topics with the term clusters generated by my system was more than that with the term clusters generated by Clusty (See Q2 in Table 5.9). There is a statistical significance in the difference between the averages of the two systems. Although the result of the pre-experiment questionnaire indicated that the participants expected multiple topics from 15 queries out of 20 queries, they found more topics by being shown multiple topics with term clusters generated by my system than those by Clusty. Additionally, the number of queries about which their knowledge was enhanced by the term clusters generated by my system exceeded those enhanced by clusters generated by Clusty (See Q3 in Table 5.9). This is because the term clusters generated by my system focused on distinguishing topics related to the query. There was also a statistical significance in the difference between the averages of the two systems.

Most participants answered that the term clusters generated by Clusty were more convenient for searching than those generated by my system (See Q4 in Table 5.10). We can say that this is because the term clusters generated by my system are hard to browse since the term clusters contain more terms than those generated by Clusty. The term clusters generated by Clusty are easier to understand than those generated by my system for general users who do not know about the query. On the other hand, most participants answered that the term clusters generated by my system were more convenient for finding interesting topics than those generated by Clusty (See Q5 in Table 5.10). This is because the term clusters generated by my system focus on distinguishing topics strongly related to the query.

5.4 Discussion

5.4.1 Query Disambiguation

I think that query ambiguity is caused by at least the following two reasons. The first is polysemy, i.e, the multiplicity of query meanings. When queries consist of a small number of terms, they are likely to refer to multiple concepts or multiple objects. The second reason is the multiplicity of the

Table 5.9: Results of Q1 to Q3. The numbers of the participants who answered Y to Q1-Q3 as for term clusters generated by Clusty and by TNG. (Average number of queries per participant is shown in parentheses.)

Question	Clusty	TNG	Statistical Significance
Q1: Did you find multiple meanings related to the query?	84 (10.5)	85 (10.6)	$t(7) = 0.16, p = n.s.$
Q2: Did you find unexpected topics related to the query?	73 (9.10)	104 (13.0)	$t(7) = 3.4, p < .01$
Q3: Did these term clusters enhance your knowledge about the query?	90 (11.3)	123 (15.4)	$t(7) = 2.6, p < .05$

Table 5.10: Number of people who answered left or right for Q4 and Q5.

Question	The left (Clusty)	The right (TNG)
Q4: Which term cluster set do you think is more convenient to search?	7	1
Q5: Which term cluster set do you think is more convenient for finding interesting topics?	1	6

Table 5.11: Comments made by four participants. Each of the comments is the contrast between the situation in which the document labels generated by Clusty are effective and the situation in which the term clusters generated by my system are effective.

Participant ID	Clusty	TNG
Participant-1	Broad search	Deep search
Participant-2	Search for definition of a word	Search for topics related to a word
Participant-3	Clear division of meanings or topics	Unbalanced division of meanings or topics
Participant-4	Search for shallow knowledge	Search for interesting topics

perspectives from which we view the concept or the object referred to by the query. We call these perspectives *facets*. Even when a query refers to a unique concept or a unique object, we can view the concept or the object from various perspectives. Therefore, the documents retrieved by such a query may provide multiple perspectives from which we can view the concept or the object referred to by the query.

1. **Multiplicity of Meanings**

   The user can recognize the ambiguity of a polysemous query when the search system presents term clusters corresponding to distinct topics related to the query (See Figure 5.5). For example, we can infer from the term clusters in Table 5.1 that the query "ジャガー" (jaguar) has multiple meanings (See Figure 5.7). Therefore, we can use these term clusters to obtain distinct search results relating to distinct meanings of the term "jaguar".

2. **Multiplicity of Facets**

   By inspecting the terms in each term cluster generated by the system, we can recognize facets, i.e., multiple perspectives from which we can view the concept or the object referred to by the query. Even when we cannot name a facet with a single label name, the combination of terms in the term cluster corresponding to the facet is enough for us to understand the corresponding perspective (See Figure 5.6). For example, the cluster of ID 3 in Table 5.2 is an enumeration of game machines. While it is difficult to give this cluster a single label name, we can understand this term cluster provides a perspective from which we can view Nintendo DS. Hearst pointed out that the facets form a hierarchical structure [Hea06a]. My current implementation of the term clustering system cannot organize the term clusters in a hierarchical manner. However, I believe the term clusters that my system provides are good candidates for the facets to be organized in a hierarchical structure.

The 20 queries I used in the experiment include queries having multiple meanings and queries having a unique meaning. The former is related to both of the ambiguities (1) and (2), and the latter is only related to the ambiguity (2) above. First, I describe the case when only the ambiguity (2) is relevant, and then I describe the case when both ambiguities are relevant.

In the retrieval experiment using term clusters described in Section 5.3.6, I tried to check the performance of Clusty and TNG without using original query terms. First, when the query has

a unique meaning, TNG's term clusters retrieve documents related to the query more efficiently than Clusty's document labels without using the original query terms. Therefore, we can say that TNG's term clusters can focus on the distinguishing topics related to the original query better than the document labels generated by Clusty. Additionally, in this case, Clusty often outputs values of facets separately in different document labels. Although the values can help in a search using the original query, we cannot recognize their roles because they are not grouped into a cluster.

Next, when the query had multiple meanings, Clusty's document labels retrieved documents related to the query more efficiently than TNG's term clusters without using it (See Section 5.3.6). Clusty often outputs a compound term that can replace the original query as the one of the meanings. Therefore, the document labels could retrieve related to the query more efficiently when the query had multiple meanings. On the other hand, TNG often outputs a set of objects or concepts. I believe that the set can be a set of values of one facet. Even if the term clusters focus on topics in each of the meanings, the topics can be related to something other than the original query when the user does not know it. In this case, the term clusters require the original query terms to focus on one of the meanings related to the query. In contrast, when the document labels and term clusters are used for the search as the additional query to the original query, the participants could find the same number of multiple meanings by using TNG's term clusters as by using Clusty's document labels (See Q1 in Table 5.9). Therefore, we can say that the term clusters generated by my system can suggest multiple meanings with the original query terms as well as the document labels generated by Clusty.

5.4.2 Discovery of Topics Related to the Query

The subjective evaluation showed that the term clusters provided by my system not only corresponded to separate topics, but also to interesting topics. According to the results of the questionnaires, six out of eight participants answered that the term clusters generated by my system helped them to find interesting topics (See Q5 in Table 5.10). Hence, we can say that this is another prominent merit of my term clustering system.

After conducting all experimental procedures, the participants were asked to answer the following two questions.

- Question 1: In which cases do you think System 1 and System 2 are more efficient than Google?
- Question 2: Which system do you think is more effective?

System 1 in Question 1 refers to Clusty and System 2 refers to my system, so the participants did not know which referred to which.

I received the following answers to Question 1: the cases in which the query has multiple meanings, the cases in which I would like to search broadly, the cases in which I do not know what kind of terms are effective for finding detailed topics relating to the query, and the cases in which I would like to have new search results based on viewpoints different from mine.

As for Question 2, four participants provided interesting answers. All of them answered that the two systems are effective in different situations, as indicated in Table 5.11. From these answers, we can conclude that term clusters given by TNG suggest more complicated structures of various topics in comparison with Clusty.

5.5 Summary

In this chapter, I compared the performance of my system using TNG with that of Clusty. I recruited eight participants to check retrieved documents by hand. I asked them to perform two different experiments. One was a retrieval experiment using term clusters generated by my system or document labels generated by Clusty without using original query terms. The other was as usability experiment of the term clusters or the document labels for query expansion. They also answered pre- and post-experiment questionnaires.

The results led us to conclude as follows for the search without using the original query. When the query has a unique meaning, TNG's term clusters retrieve documents related to the query more efficiently than Clusty's document labels. On the other hand, when the query has multiple meanings, Clusty's document labels retrieve documents related to the query more efficiently than TNG's term clusters. However, when the document labels and term clusters are used for the search as the additional query to the original query, the participants could find the same number of the multiple meanings by using TNG's term clusters as by using Clusty's document labels. Therefore, we can

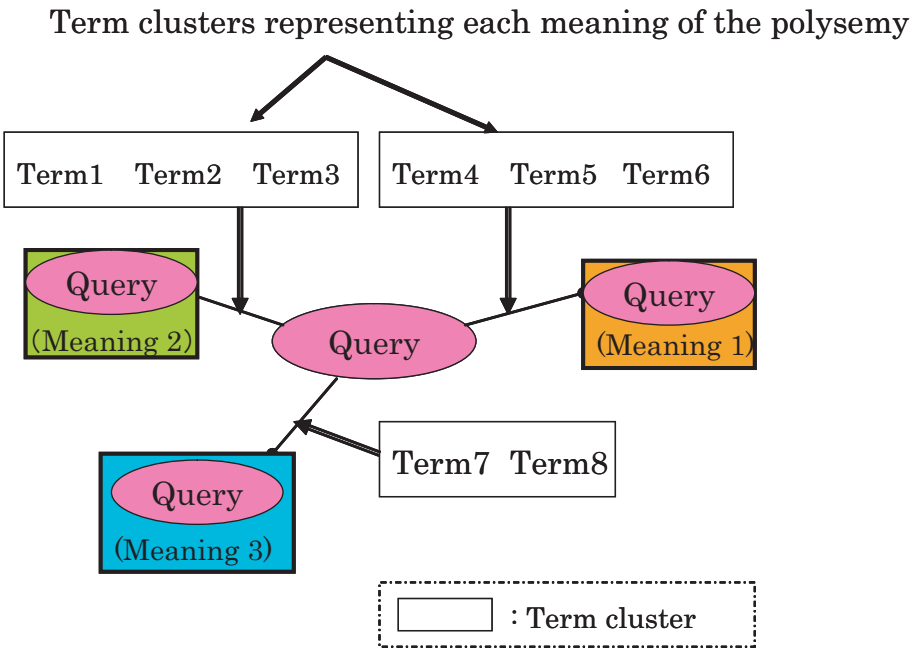


Figure 5.5: Example of case in which the term clusters help to find polysemy.

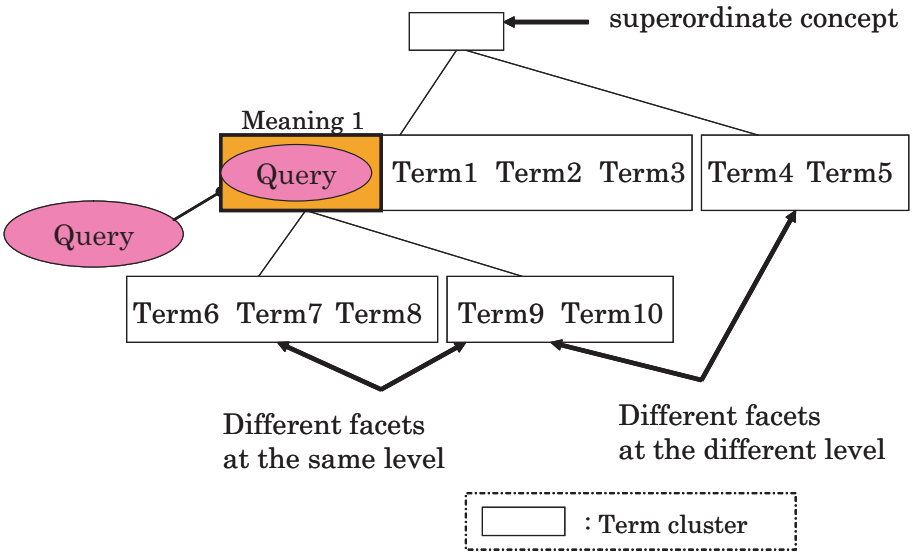


Figure 5.6: Example of case in which the term clusters help to find facets.

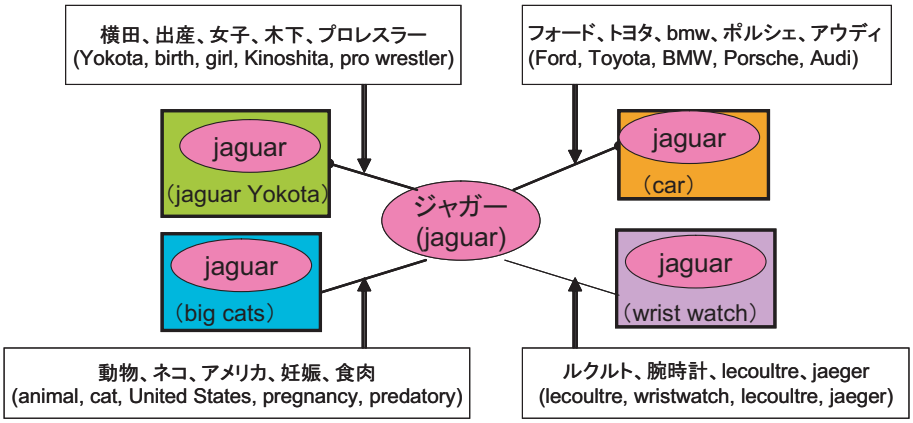


Figure 5.7: Representation of multiple meanings of the query “ジャガー” (jaguar).

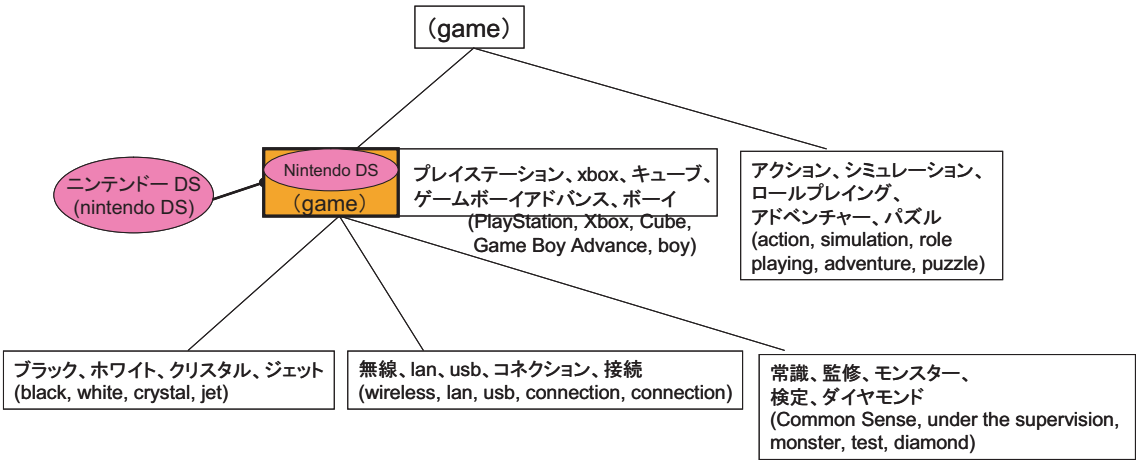


Figure 5.8: Representation of facets for the query “ニンテンドー DS” (Nintendo DS).

say that the term clusters generated by my system suggest multiple meanings with the original query terms as well as do the document labels generated by Clusty.

As for the usability experiment, we can say that we can easily recognize multiple meanings of the query by showing multiple topics related to the query because the number of words recognized as polysemic words increased. Moreover, the number of queries to which the participants found unexpected topics with the term clusters generated by my system is more than that with the term clusters generated by Clusty. Furthermore, most participants answered that the term clusters generated by my system were more convenient for finding interesting topics than those generated by Clusty. Therefore, we can say that the term clusters generated by my system can focus on the topics related to the original query.

I think that query ambiguity is caused by at least the following two reasons. The first is polysemy, i.e, the multiplicity of query meanings. The second reason is the multiplicity of the perspectives, i.e., facets, from which we view the concept or the object referred to by the query. As for the former, we can use the term clusters generated by my system to obtain distinct search results relating to distinct meanings of the original query. As for the latter, I believe that the term clusters that my system provides are good candidates for the facets to be organized in a hierarchical structure.

Chapter 6

Conclusion

In this study, I aim to represent multiple topics related to a query for searching. The search results are worthful for seeking knowledge because there are numerous kinds of contents from numerous information sources at any time. However, the information from the search results is not well organized like books having a table of contents. We usually enter only a few terms to form the query. Even if we use effective query terms to focus on a unique meaning, various topics or perspectives related to the query can be contained in the search results. As a result, existing search engines often return a long list of search results. This is because the search engines cannot narrow the search results automatically, since all the documents are related to the query and these documents can be fit our needs. Therefore, I tried to represent multiple topics related to the query and to observe the solution for query ambiguities.

First, I proposed a method to extract topical terms and generated term clusters using the extracted terms. I assume that a term co-occuring frequently with a specific set of terms is distinguishing. After proposing the initial formulae called TNG1 and TNG2, I proposed a sophisticated formula called TNG. I generated term clusters based on the distributional clustering algorithm.

Next, I examined the proposed method through three kinds of experiments. One is the experiment of the term extraction for query expantison by using TNG1 and TNG2. As a result, TNG1 and TNG2 realized good average precisions for the query expansion. In addition, many of the extracted terms were related to more specific topics than that implied by the original ambiguous query terms. Another is the experiment of the topical term extraction by using TNG in comparison with other term weighting methods on multiple data sets. The results showed that TNG can extract terms

strongly related to any one of several topics contained in the document set. The other is the experiment of the topical term clustering for query refinement by using TNG in comparison with other term weighting methods on multiple data sets. The results showed that TNG is an efficient term weighting method for detection of topics included in a heterogeneous set of documents. I think that TNG can be used for query expansion that considers that various topics are contained in the first-retrieved documents and that users can select one of those topics efficiently with my method.

Finally, I developed a system using TNG to show topical term clusters that can be used for query refinement. I also examined TNG in comparison with Clusty. I think that query ambiguity is caused by at least the following two reasons. The first is polysemy, i.e, the multiplicity of query meanings. The second reason is the multiplicity of the perspectives, i.e., facets, from which we view the concept or the object referred to by the query. As a result of the experiments, I observed the following two solutions. As for the polysemy, we can use the term clusters generated by my system to obtain distinct search results relating to distinct multiple meanings of the original query. As for the facets, I believe that the term clusters that my system provides are good candidates for the facets to be organized in a hierarchical structure.

Bibliography

[Adk05] H. P. Adkisson. Use of faceted classification, 2005. http://www.webdesignpractices.com/navigation/facets.html.

[BM98] L. Douglas Baker and Andrew K. McCallum. Distributional clustering of words for text classification. In *Proceedings of SIGIR-98*, pp. 96–103, 1998.

[BYRN99] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[CCK+03] K. Chen, H. Chen, N. Kando, K. Kuriyama, S. Lee, S. H. Myaeng, K. Kishida, K. Eguchi, and H. Kim. Overview of clir task at the third ntcir workshop, 2003.

[CG95] K. Church and W. Gale. Inverse document frequency (IDF): A measure of deviations from poisson. In *Proc. of 3rd Workshop on Very Large Corpora*, pp. 121–130, 1995.

[Cha02] S. Chakrabarti. *Mining the Web:Discovering Knowledge from Hypertext Data*. Morgan-Kaufmann Publishers, 2002.

[CKPT92] D.R. Cutting, D.R. Karger, J.O. Pedersen, and J.W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proc. of SIGIR'92*, pp. 318–329, 1992.

[DDI06] W. Dakka, R. Dayal, and P.G. Ipeirotis. Automatic discovery of useful facet terms. In *Proc. of the ACM SIGIR 2006 Workshop on Faceted Search*, 2006.

[Den03] W. Denton. How to make a faceted classification and put it on the web, 2003. http://www.miskatonic.org/library/facet-web-howto.html.

[DH02] C. Ding and X. He. Cluster merging and splitting in hierarchical clustering algorithms. In *IEEE International Conference on Data Mining (ICDM'02)*, pp. 139–146, 2002.

[DHZ+01] C. Ding, X. He, H. Zha, M . Gu, and H. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proc. of IEEE Int'l Conf. Data Mining*, pp. 107–114, 2001.

[EOAI04] K. Eguchi, K. Oyama, A. Aizawa, and H. Ishikawa. Overview of web task at the fourth ntcir workshop, 2004.

[EOI+03] K. Eguchi, K. Oyama, E. Ishida, N. Kando, and K. Kuriyama. Overview of the web retrieval task at the third ntcir workshop, 2003.

[FDNY+06] E. A. Fox, F. Das-Neves, X. Yu, R. Shen, S. Kim, and S. Fan. Exploring the computing literature with visualization and stepping stones & pathways. *Communications of the ACM*, Vol. 49, No. 4, pp. 52–58, 2006.

[FTZ] H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. In *Proc. of SIGIR 2004*.

[Hea06a] M. A. Hearst. Clustering versus faceted categories for information exploration. *Communications of the ACM*, Vol. 49, pp. 59–61, 2006.

[Hea06b] M. A. Hearst. Design recommendations for hierarchical faceted search interfaces. In *Proc. of the ACM SIGIR 2006 Workshop on Faceted Search*, 2006.

[HLF05] H. Huo, J. Liu, and B. Feng. Multinomial approach and multiple-bernoulli approach for information retrieval based on language modeling. In *FSKD (1)*, pp. 580–583, 2005.

[HNN+00] T. Hisamitsu, Y. Niwa, S. Nishioka, H. Sakurai, O. Imaichi, M. Iwayama, and A. Takano. Extracting terms by a combination of term frequency and a measure of term representativeness. *International journal of theoretical and applied aissues in specialized communication*, Vol. 6, No. 2, pp. 211–232, 2000.

[HP96] M. Hearst and J. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proc. of SIGIR'96*, pp. 76–84, 1996.

[HSC06] M. A. Hearst, P. Smalley, and C. Chandler. Faceted metadata for information architecture and search. In *CHI 2006 Course*, 2006.

[HWA06] Atsuhiro Takasu Hiromi Wakaki, Tomonari Masada and Jun Adachi. A new measure for query disambiguation using term co-occurrences. In *Proc. of 7th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, pp. 904–911, 2006.

[ipa] ipadic-2.5.1. http://chasen.naist.jp/stable/ipadic/.

[ISDK03] S. Mallela I. S. Dhillon and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classi cation. *Journal of Machine Learning Research(JMLR): Special Issue on Variable and Feature Selection*, pp. 1265–1287, 2003.

[JCSB02] H. Joho, C. Coverson, M. Sanderson, and M. Beaulieu. Hierarchical presentation of expansion terms. In *Proc. of SAC'02*, pp. 645–649, Madrid, Spain, 2002.

[JSBS98] B.J. Jansen, A. Spink, J. Bateman, and T. Saracevic. Searchers, the subjects they search, and sufficiency: A study of a large sample of excite searches. In *1998 World Conference on the WWW and Internet*, 1998.

[LC00] D.J. Lawrie and W.B. Croft. Discovering and comparing hierarchies. In *Proc. of RIAO 2000*, pp. 314–330, 2000.

[Lee] L. Lee. Measures of distributional similarity. In *Proc. of 27th Annual Meeting of the Association for Computational Linguistics (ACL 1999)*, pp. 25–32.

[Mak05] M. Maki. Findex: Search result categories help users when document ranking fails. In *Proc. of the SIGCHI conference on Human factors in computing systems*, pp. 131–140, 2005.

[MeC] MeCab. http://mecab.sourceforge.jp/.

[MI04] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-ocurrence statistical information. *International Journal on Artificial Intelligence Tools*, Vol. 13, pp. 157–169, 2004.

[Mor05] P. Morville. *Ambient Findability*. O'Reilly Media, 2005.

[NIH+99] Y. Niwa, M. Iwayama, T. Hisamitsu, S. Nishioka, A. Takano, H. Sakurai, and O. Imaichi. Interactive document search with *DualNAVI*. In *Proc. of NTCIR'99*, pp. 123–130, 1999.

[OKI04] S. Oyama, T. Kokubo, and T. Ishida. Domain-specific web search with keyword spices. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, pp. 17–27, Jan 2004.

[OKIY01] S. Oyama, T. Kokubo, T. Ishida, and T. Yamada. Keyword spices: A new method for building domain-specific web search engines. In *Proc. 17th Int'l Joint Conf. Artificial Intelligence (IJCAI-01)*, pp. 1457–1463, 2001.

[one] Less people use 1 word phrase in search engines according to onestat.com. http://www.onestat.com/html/aboutus_pressbox45-search-phrases.html.

[ONY98] Y. Ohsawa, E.B. Nels, and M. Yachida. Keygraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In *Proc. of IEEE ADL'98*, 1998.

[PF00] W. Pratte and L. Fagan. The usefulness of dynamically categorization search results. *Journal of the American Medical Informatics Association*, Vol. 7, pp. 605–617, 2000.

[RJ04] K. A. Ross and A. Janevski. Querying faceted databases. In *Proc. of the Second Workshop on Semantic Web and Databases*, pp. 199–218, 2004.

[RJ05] J. Rennie and T. Jaakkola. Using term informativeness for named entity detection. In *Proc. of SIGIR'05*, pp. 353–360, 2005.

[Rob90] S. E. Robertson. On term selection for query expansion. *Journal of Documentation*, Vol. 46, No. 4, pp. 359–364, 1990.

[RW99] S.E. Robertson and S. Walker. Okapi/keenbow at trec-8. In *TREC*, 1999.

[SB90] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, Vol. 44, No. 4, pp. 288–297, 1990.

[SC99] M. Sanderson and B. Croft. Deriving concept hierarchies from text. In *Proc. of SIGIR 99*, pp. 206–213, 1999.

[Sea06] SearchTools.com. Faceted metadata search and browse, 2006. http://searchtools.com/info/faceted-metadata.html.

[Seb02] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, Vol. 34, No. 1, pp. 1–47, 2002.

[TNN+] A. Takano, Y. Niwa, S. Nishioka, M. Iwayama, T. Hisamitsu, O. Imaichi, and H. Sakurai. Associative information access using dualnavi. In *Proc. of ICDL'00*, pp. 285–289.

[TNN+00] A. Takano, Y. Niwa, S. Nishioka, M. Iwayama, T. Hisamitsu, O. Imaichi, and H. Sakurai. Information access based on associative calculation. In *Proc. of SOFSEM2000*, pp. 187–201, 2000.

[TRE] TREC. trec_eval, http://trec.nist.gov/trec_eval/.

[Tun06] D. Tunkelang. Dynamic category sets: An approach for faceted search. In *Proc. of the ACM SIGIR 2006 Workshop on Faceted Search*, 2006.

[YDNF05] X. Yu, F. Das-Neves, and E.A. Fox. Hard queries can be addressed with query splitting plus stepping stones and pathways. *Bulletin of the IEEE-CS Technical Committee on Data Engineering*, Vol. 28, No. 4, pp. 29–38, 2005.

[YH04] M. Yoshioka and M. Haraguchi. Study on the combination of probabilistic and boolean ir models for www documents retrieval. In *Working Notes of NTCIR-4 (Supplement Volume)*, pp. 9–16, 2004.

[YP97] Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization. In *Proc. of ICML-97*, pp. 412–420, 1997.

[YSLH03] K. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In *Proc. of CHI 2003*, pp. 401–408, 2003.

[ZE99] O. Zaimir and O. Etzioni. Grouper: A dynamic clustering interface to Web search results. In *Proc. of WWW8*, pp. 1361–1374, 1999.

[若木 05] 若木裕美, 正田備也, 高須淳宏, 安達淳. 検索語の曖昧性を解消するキーワードの提示手法. *DBSJ Letters*, Vol. 4, No. 2, pp. 41–44, 2005.

[若木 06a] 若木裕美, 正田備也, 高須淳宏, 安達淳. 具体性指向単語クラスタリングによる網羅的トピックの発見と検索質問拡張支援. 電子情報通信学会第 17 回データ工学ワークショップ (DEWS2006), pp. 2C–i4, 2006.

[若木 06b] 若木裕美, 正田備也, 高須淳宏, 安達淳. 検索語の曖昧性解消のためのトピック指向単語抽出および単語クラスタリング. 情報処理学会論文誌データベース (TOD), Vol. 47, No. SIG19(TOD32), pp. 72–85, 2006.

[小久 02] 小久保卓, 小山聡, 山田晃弘, 北村泰彦, 石田亨. 検索隠し味を用いた専門検索エンジンの構築. 情報処理学会論文誌, Vol. 43, pp. 1804–1813, June 2002.

[松岡 05] 松岡正剛. 検索ビジネス最前線. *Internet magazine*, No. 129, pp. 26–53, 10 2005.

[松尾 02] 松尾豊, 石塚満. 語の共起の統計情報に基づく文書からのキーワード抽出アルゴリズム. 人工知能学会論文誌, Vol. 17, pp. 213–227, 2002.

[大澤 99] 大澤幸生, ネルス E. ベンソン, 谷内田正彦. Keygraph: 語の共起グラフの分割・統合によるキーワード抽出. 電子情報通信学会論文誌, Vol. J82-D-I, pp. 391–400, 2 1999.

Publication Lists

Refereed Journal Articles

[1] Hiromi Wakaki, Tomonari Masada, Atsuhiro Takasu, and Jun Adachi, "Query Refinement based on Comprehensive Representation of Multiple Topics",*ACM Transactions on Asian Language Information Processing*. (Submitted)

[2] 若木裕美, 正田備也, 高須淳宏, 安達淳, "検索語の曖昧性解消のためのトピック指向単語抽出および単語クラスタリング", **情報処理学会論文誌データベース (TOD)**, 47(SIG19(TOD32)):72–85, 2006.

[3] 若木裕美, 正田備也, 高須淳宏, 安達淳, "検索語の曖昧性を解消するキーワードの提示手法", **DBSJ Letters**, 4(2):41–44, 2005.

Conference Proceedings

[4] Hiromi Wakaki, Tomonari Masada, Atsuhiro Takasu, and Jun Adachi, "Query Refinement based on Topical Term Clustering", In *Proc. of RIAO 2007*, 2007. (Submitted)

[5] Hiromi Wakaki, Tomonari Masada, Atsuhiro Takasu, and Jun Adachi, "A New Measure for Query Disambiguation using Term Co-occurrences", In *Proc. of 7th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, Spain, Sep. 20-23, pages. 904-911, 2006.

[6] Hiromi Wakaki and Hitoshi Iba, "AVICE: Evolving Avatar's Movernent", In *Proc. of Genetic and Evolutionary Computation Conference (GECCO-2003)*, Chicago, USA, pages. 1816–1817, 2003.

[7] Hiromi Wakaki and Hitoshi Iba, "AVICE: Evolving Avatar's Movements", In *Proc. of 10th International Conference on Human-Computer Interaction (HCI2003)*, Crete, Greece, pages. 540–549, 2003.

[8] Hiromi Wakaki, Nao Tokui and Hitoshi Iba, "Motion design of a 3D-CG avatar using interactive evolutionary computation", In *Proc. of 2002 IEEE international Conference on Systems, man and Cybernetics (SMC'02)*, Hammamet, Tunisia, IEEE Press, 2002.

[9] Hitoshi Iba, Nao Tokui, and Hiromi Wakaki, "3D-CG Avatar Motion Design by means of Interactive Evolutionary Computation", In *Proc. of Hybrid Intelligent Systems (HIS2002)*, pp.540-549, Santiago, Chile, December. 1-4, 2002.

[10] Hiromi Wakaki and Htoshi Iba, "Motion design of a 3D-CG avatar that uses humanoid animation", In *Proc. of the 4th International Workshop on Emergent Synthesis (IWES'02)*, Kobe University, Japan, 2002.

Research Reports

[11] 若木裕美, 正田備也, 高須淳宏, 安達淳, “トピック指向単語クラスタリングを用いた複数トピックの包括的提示による検索支援”, **電子情報通信学会第 18 回データ工学ワークショップ (DEWS2007)**, 2007. (Submitted)

[12] 若木裕美, 正田備也, 高須淳宏, 安達淳, “具体性指向単語クラスタリングによる網羅的トピックの発見と検索質問拡張支援”, **電子情報通信学会第 17 回データ工学ワークショップ (DEWS2006)**, 2C-i4, 沖縄, 2006.

[13] 若木裕美, 正田備也, 高須淳宏, 安達淳, “検索語の曖昧性を解消するキーワードの提示手法”, **情報処理学会研究報告「データベースシステム」**, 137:269–276, 2005.

[14] 若木裕美, 伊庭斉志, “AVICE: 進化論的手法による動作作成の支援システム”, 第 9 回 MPS シンポジウム (進化的計算シンポジウム), **情報処理学会シンポジウムシリーズ**, Vol2003, No.2, pages. 177–180, 2003.

[15] 若木裕美, 伊庭斉志, “対話型進化計算法を用いた 3D Avatar 設計”, **電子情報通信学会総合大会予稿集**, 2002.

Books

[16] 若木裕美, 伊庭斉志, “対話型進化計算法による３次元アニメーションの生成”, Computer Today, pages. 72–78, 2002, Nov, No.112.

Honors

Database Workshop 2005 (DBWS2005) 学生研究発表奨励賞